



TUGAS AKHIR - TE 145561

**RANCANG BANGUN ROBOT PEMBUKA BRANKAS DENGAN
SISTEM PENGIRIMAN KODE PUTAR MELALUI SMS *GATEWAY***

Shofia Kholisatus Sa'adah
NRP. 2212 039 002
Alin Wahyu Setiawan
NRP. 2212 039 043

Dosen Pembimbing
Rudy Dikairono, S.T., M.T
Onie Meiyanto, S.Pd

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 145561

***THE SAFECRACKING ROBOT DESIGN WITH COMBINATION
CODE DELIVERY SYSTEM USING SMS GATEWAY***

Shofia Kholisatus Sa'adah
NRP. 2212 039 002
Alin Wahyu Setiawan
NRP. 2212 039 043

Lecturer Counsellor
Rudy Dikairono, S.T., M.T
Onie Meiyanto, S.Pd

Electrical Engineering D3 Study Program
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

RANCANG BANGUN ROBOT PEMBUKA BRANKAS DENGAN SISTEM PENGIRIMAN KODE PUTAR MELALUI SMS GATEWAY


TUGAS AKHIR

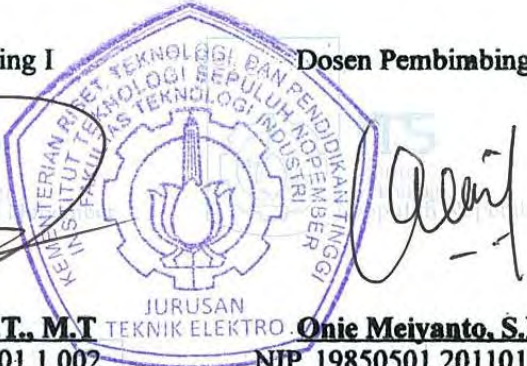
**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Elektro Industri
Program Studi D3 Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II


Rudy Dikairono, S.T., M.T.
NIP. 19810325 200501 1 002


Onie Meiyanto, S.Pd
NIP. 19850501 201101 1 008

**SURABAYA
JUNI, 2015**

RANCANG BANGUN ROBOT PEMBUKA BRANKAS DENGAN SISTEM PENGIRIMAN KODE PUTAR MELALUI MSGATEWAY

Nama Mahasiswa : Shofia Kholisatus Sa'adah
NRP : 2212 039 002
Nama Mahasiswa : Alin Wahyu Setiawan
NRP : 2212 039 043
Dosen Pembimbing 1 : Rudy Dikairono, S.T., M.T
NIP : 19810325 200501 1 002
Dosen Pembimbing 2 : Onie Meiyanto, S.Pd
NIP : 19850501 201101 1 008

ABSTRAK

Suatu perusahaan pasti memiliki tempat penyimpanan khusus untuk menyimpan semua aset pentingnya. Salah satunya dengan menyimpannya dalam brankas. Dapat dipastikan suatu perusahaan besar memiliki lebih dari satu buah brankas. Yang mana untuk menjaga keamanan brankas tersebut, kode putar yang digunakan haruslah berbeda – beda. Dengan permasalahan pemegang hak akses brankas yang memiliki jadwal sangat padat dan tidak selalu berada di tempat, maka tugas akhir ini menjadi solusinya.

Robot pembuka brankas dengan sistem pengiriman kode putar melalui SMS Gateway. Robot ini dapat menerima SMS (*Short Message Service*) berupa kode putar dari pemilik brankas. Kemudian SMS yang diterima oleh robot tersebut dikonversikan menjadi putaran pada motor planetary yang nantinya robot tersebut memutar putaran kode putar brankas dan brankas akan dapat terbuka.

Dengan beberapa pengujian yang telah dilakukan baik pengujian secara mekanik maupun elektrik, robot pembuka brankas ini mampu membuka kunci brankas putar sesuai dengan kode yang telah dikirimkan oleh pemilik brankas melalui SMS. Selain itu, robot ini dapat mengirimkan SMS konfirmasi kepada pemilik brankas apabila brankas telah terbuka ataupun ketika terjadi *error* saat robot sedang membuka brankas.

Kata Kunci : Brankas, *Rotary Encoder*, SMS Gateway.

-----Halaman ini sengaja dikosongkan-----

THE SAFECRACKING ROBOT DESIGN WITH COMBINATION CODE DELIVERY SYSTEM USING SMS GATEWAY

Student Name : Shofia Kholisatus Sa'adah
ID Number : 2212039 002
Student Name : Alin Wahyu Setiawan
ID Number : 2212 039 043
Supervisor 1 : Rudy Dikairono, S.T., M.T
ID Number : 19810325 200501 1 002
Supervisor 2 : Onie Meiyanto, S.Pd
ID Number : 19850501 201101 1 008

ABSTRACT

Every company absolutely have a secret place to save all of the important assets. One of them, to save into the safe box. Ascertained, a big company have more than one. Saving security of the safe box, rotate code must be different each other. With the problem, the owner of the safe box have a busy schedule and not always in certain place, so the final project is a solution.

The Safecracking Robot Design with Combination Code Delivery System Using SMS Gateway. *This robot can receive SMS (Short Message Service) that contain rotate code from the owner of the safebox. And then, the SMS that received by Robot converted to stepper motor or planetary rotation. So, the robot able to rotate the rotation of rotate code and then the safe will be open.*

With some trials have been done include mechanical and electrical trials, The Safecracking Robot can open the combination code of the safe which is suitable with the code that has been sent by a message. Otherwise, The robot enable to send confirmation message when the safe can be opened safely and when the system is error.

Keyword : *Safe box, Rotary Encoder, SMS Gateway.*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada junjungan kita Nabi Besar Muhammad SAW.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma pada Bidang Studi Elektro Industri, Program Studi D3 Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul :

"RANCANG BANGUN ROBOT PEMBUKA BRANKAS DENGAN SISTEM PENGIRIMAN KODE PUTAR MELALUI SMS GATEWAY"

Dengan terselesaikannya Tugas Akhir ini, Penulis menyampaikan terima kasih yang sebesar - besarnya kepada :

1. Kedua orang tua yang senantiasa mendo'akan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Rudy Dikairono, S.T., M.T. selaku dosen pembimbing.
3. Bapak Onie Meiyanto, S.Pd selaku dosen pembimbing dari BLKIP.
4. Bapak Chong Konggo yang telah memberikan dukungan secara materi.
5. Teman - teman Elektro Industri Angkatan 2012 yang selalu memberikan doa, semangat, dan dukungannya.
6. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juni 2015

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

JUDUL	i
LEMBAR PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Perencanaan	2
1.5 Sistematika Laporan Tugas Akhir	2
1.6 Relevansi	3

BAB II TEORI PENUNJANG

2.1 Brankas	5
2.2 Mikrokontroler ATmega 128	6
2.3 <i>Rotary Encoder</i>	7
2.4 <i>GSM Shield</i> (Modem GSM SIM 900 A)	9
2.5 <i>GearBox</i>	10
2.6 DC Motor <i>Driver</i>	11
2.7 LCD 8 x 2	12
2.8 <i>Reed Switch</i>	12
2.9 <i>Limit Switch</i>	13

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Blok Diagram Fungsional	15
3.2 Perancangan Perangkat Keras (<i>Hardware</i>)	16
3.2.1 Perancangan Elektronik	17
3.2.1.1 Skematik Mikrokontroler ATmega 128 ...	17
3.2.1.2 Pengkoneksian Modul GSM SIM 900A ...	18
3.2.1.3 Pengendalian DC Planet <i>Geared</i> Motor PG36 Menggunakan Modul Driver Motor	20

3.2.1.4	Pengkoneksian <i>Voltage Regulator</i>	20
3.2.1.5	Perancangan Sensor <i>Limit Switch</i> dan <i>Reed Switch</i>	21
3.2.2	Perancangan Mekanik	22
3.2.2.1	Perancangan Pemegang <i>Knob</i>	23
3.2.2.2	Perancangan <i>Casing</i> Elektronik	25
3.2.2.3	Perancangan Bagian Inti Robot	28
3.3	Perancangan Perangkat Lunak (<i>Software</i>)	35
3.3.1	Pemograman Mikrokontroler ATmega 128	35
3.3.2	Pemograman DC Planet <i>Geared</i> Motor	37
3.3.3	Pemograman Modul GSM SIM 900A	38
3.3.4	Pemograman Tampilan LCD	39

BAB IV PENGUJIAN DAN ANALISA DATA

4.1	Pengujian Modul GSM SIM 900A	41
4.2	Pengujian <i>Driver</i> Motor	43
4.3	Pengujian Program Pembacaan <i>Rotary Encoder</i> pada DC Planet <i>Geared</i> Motor	44
4.4	Pengujian Program <i>Driver</i> Motor terhadap DC Planet <i>Geared</i> Motor dan Dibaca oleh Kontroler	48
4.5	Pengujian Program Modul GSM SIM 900A terhadap Kontroler	50
4.6	Pengujian Kepresisian Putaran Motor	53
4.7	Pengujian Alat Secara Keseluruhan	56

BAB V PENUTUP

5.1	Kesimpulan	61
5.2	Saran	61

DAFTAR PUSTAKA	63
-----------------------------	----

LAMPIRAN A GAMBAR	A-1
--------------------------------	-----

LAMPIRAN B PROGRAM	B-1
---------------------------------	-----

LAMPIRAN C DATASHEET	C-1
-----------------------------------	-----

DAFTAR RIWAYAT HIDUP	D-1
-----------------------------------	-----

DAFTAR TABEL

Tabel 2.1	Diagram <i>State Encoder Incremental</i>	8
Tabel 2.2	Fungsi Perintah AT pada SIM 900A	9
Tabel 3.1	PIN Mikrokontroler ATmega 128 yang dipergunakan	18
Tabel 4.1	Hasil Pengujian <i>Driver Motor</i>	43
Tabel 4.2	Hasil Pengujian Program <i>Rotary Encoder</i>	46
Tabel 4.3	Hasil Pengujian Program <i>Driver Motor</i>	49
Tabel 4.4	Hasil Pengujian Program Modul GSM SIM 900A	53
Tabel 4.5	Hasil Pengujian Kepresisian Putaran Motor	54
Tabel 4.6	Hasil Pengujian Alat Secara Keseluruhan	59
Tabel 4.7	Hasil Waktu Pengujian Alat Secara Keseluruhan	60

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Suatu perusahaan pasti memiliki tempat penyimpanan khusus untuk menyimpan aset - aset pentingnya. Salah satu tempat penyimpanan khusus tersebut adalah brankas. Dapat dipastikan suatu perusahaan besar memiliki lebih dari satu buah brankas untuk menyimpan aset penting perusahaan tersebut.

Memiliki brankas dengan jumlah yang banyak pasti memerlukan sistem keamanan yang sangat besar pula. Untuk menjaga keamanan brankas tersebut salah satunya dengan menggunakan kode putar yang berbeda – beda pada setiap brankasnya. Membuka brankas tersebutpun tidak semua karyawan perusahaan memiliki akses. Hanya orang – orang kepercayaan sajalah yang dapat membukanya. Dengan permasalahan pemegang hak akses memiliki jadwal kegiatan yang sangat padat dan tidak berada di tempat, sedangkan aset penting dalam brankas sangat diperlukan oleh perusahaan.

Maka dari itu, perlu direalisasikan suatu robot yang dapat dipercaya oleh pemegang hak akses brankas untuk membuka kunci brankasnya. Dan hal tersebutlah yang menjadi dasar pembuatan tugas akhir ini, yaitu suatu robot yang dapat menerima kode putar brankas melalui SMS (*Short Message Service*) yang kemudian robot ini akan memutar putaran brankas tersebut dan brankas dapat dibuka.

1.2 Rumusan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini adalah sebagai berikut :

- Tidak adanya teknologi yang dapat membuka brankas manual dimana kode brankas tersebut dapat dikirim melalui media SMS (*Short Message Service*).
- Tidak selalu pemilik brankas *standby* di tempat untuk membuka brankas saat ada aset yang dibutuhkan oleh perusahaan.

1.3 Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari tugas akhir ini adalah :

- Robot hanya dapat membuka kunci brankas *type* BERNSTEIN size 2
- Tugas Akhir ini hanya dapat membuka brankas dengan kode putar yang dikirim melalui SMS.
- Pengontrolan robot ini menggunakan mikrokontroler ATmega 128 dan menggunakan Modul GSM SIM 900A.
- Sensor yang digunakan sebagai pembaca putaran motor dan juga pembaca indeks kode putar pada brankas adalah sensor *rotary encoder* yang mana sensor tersebut terdapat pada DC Planet *Geared* Motor Type PG36 itu sendiri.

1.4 Tujuan

Tujuan kami membuat Tugas Akhir ini adalah sebagai berikut :

- Merancang dan membuat suatu robot yang dapat membuka brankas manual dimana kode brankas tersebut dapat dikirim melalui media SMS (*Short Message Service*).
- Memberi kemudahan kepada pemilik brankas yang ingin membuka brankasnya tanpa diketahui kode brankasnya dan dapat dilakukan dari jarak jauh.

1.5 Sistematika

Sistematika pembahasan tugas akhir ini terdiri dari lima bab, yaitu Pendahuluan, Teori Penunjang, Perancangan dan Pembuatan Alat, Pengujian dan Analisa Alat, serta Penutup.

BAB I : PENDAHULUAN

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

BAB II : TEORI PENUNJANG

Membahas tentang teori - teori penunjang yang diperlukan dan dipergunakan sebagai penunjang pengerjaan Tugas Akhir

Bab III : PERANCANGAN DAN PEMBUATAN ALAT

Membahas tentang perencanaan dan pembuatan perangkat keras (*hardware*) yang terdiri dari perancangan elektronik dan perancangan mekanik

serta pembuatan dan perancangan perangkat lunak (*software*).

BAB IV : PENGUJIAN DAN ANALISA ALAT

Membahas tentang pengujian alat yang terdiri dan pengujian perangkat keras dan juga perangkat lunak. Begitu pula dengan pengukuran.

BAB V : PENUTUP

Menjelaskan tentang kesimpulan dari tugas akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.6 Relevansi

- Mempermudah pemilik brankas untuk dapat membuka brankasnya tanpa perlu datang langsung untuk membuka brankasnya dan tanpa perlu khawatir kode putar brankasnya diketahui oleh orang lain.
- Menghemat waktu pemilik brankas

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI PENUNJANG

Pada bab ini dibahas mengenai teori - teori yang menunjang dalam proses pembuatan alat Tugas Akhir. Teori yang menunjang dalam Tugas Akhir ini diantaranya adalah Brankas, Mikrokontroler ATmega 128, Rotary Encoder, Modul GSM SIM 900A, *DC Planet Geared Motor*, *DC Gearbox*, DC Motor Driver, dan LCD 2x16, *Reed Switch*, dan *Limit Switch*.

2.1 Brankas [1]

Brankas adalah lemari atau kotak besi tahan api dipergunakan untuk melindungi barang - barang berharga dari bahaya kebakaran dan pencurian/perampokan (seperti uang, surat - surat berharga, perhiasan, dan lain - lain). Kata brankas berasal dari Bahasa Belanda yaitu "*braden*" yang artinya membakar dan "*kast*" yang artinya lemari. Jadi artinya adalah lemari tahan kebakaran. Sedangkan dalam Bahasa Indonesia yang berarti "lemari besi" yaitu lemari yang terbuat dari besi. Brankas biasanya berbentuk kubus berongga atau silinder dengan satu sisi yang dapat dilepas atau berengsel yang berfungsi sebagai pintu.

Berdasarkan ketahanannya brankas di bagi menjadi 3 jenis, yaitu :

1. *Steel Safes* (brankas ini terbuat dari plat besi/baja). Brankas ini tidak tahan api dan biasanya hanya di pergunakan untuk penyimpanan uang yang relatif sedikit dan dipergunakan untuk kebutuhan operasional usaha skala kecil.
2. *Fire Safes* (brankas ini sudah dilengkapi dengan fitur tahan api). Brankas ini terlihat dari ketebalan *body* brankas dan apabila diketuk terasa ada isi dari *body* brankas terdapat.
3. *Fire and Burglar Safes* (Brankas ini merupakan brankas yang tahan api dan juga tahan dobrak). Brankas jenis ini sangat di rekomendasikan penggunaannya karena sudah memiliki keamanan yang relatif tinggi. Brankas ini biasanya juga sudah dilengkapi dengan fitur *relocking* (penguncian otomatis). Dilihat dari bentuk fisiknya, brankas ini ber dinding tebal, bahan platnya juga lebih tebal dari jenis brankas tahan api.

Kunci pengaman pada brankas berdasarkan sistem kerjanya ada 2 macam. Yaitu digital dan analog. Tidak jarang kedua sistem tersebut disatukan, maksudnya brankas yang bersistem digital di sertai dengan kunci analog (kunci manual).

Tipe Brankas yang dipergunakan pada Tugas Akhir ini adalah Lemari Pengaman Tahan Api *type* BERNSTEIN size 2 dengan ukuran luar 64 x 52 x 50 (cm) dan ukuran dalam 49 x 37 x 32 (cm) dengan berat 285 kg.

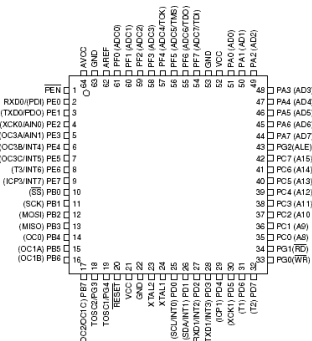
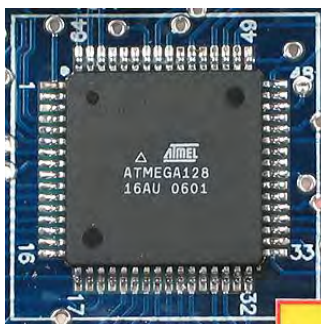
Berikut ini adalah gambar brankas yang dipergunakan :



Gambar 2.1. Brankas

2.2 Mikrokontroler ATmega 128 [2]

Mikrokontroler ATmega 128 merupakan mikrokontroler AVR yang mempunyai kapasitas *flash* memori terbesar diantara *type* lainnya. Yaitu sebesar 128 KB.



Gambar 2.2 Mikrokontroler ATmega 128

Mikrokontroler AVR ATmega 128 memiliki spesifikasi sebagai berikut :

- Saluran I/O sebanyak 56 buah, yaitu *Port A*, *Port B*, *Port C*, *Port D*, *Port E*, *Port F* dan *Port G*.
- ADC 10 bit sebanyak 8 saluran.
- 2 buah *Timer/Counter* 8 bit dan 2 buah *Timer/Counter* 16 bit.
- Dua buah PWM 8 bit.
- *Watchdog Timer* dengan osilator internal.
- Internal SRAM sebesar 4 kbyte.
- Memori *flash* sebesar 128 kbytes.
- Interupsi Eksternal.
- *Port* antarmuka SPI.
- EEPROM sebesar 4 kbyte.
- *Real time counter*.
- 2 buah *Port* USART untuk komunikasi serial.
- 6 kanal PWM.
- Tegangan operasi sekitar 4,5 V sampai dengan 5,5V.

2.3 Rotary Encoder [3][4]

Rotary Encoder disebut juga *encoder* poros, adalah perangkat elektro-mekanis yang mengubah sudut posisi/gerakan poros atau mengubah poros ke kode analog ataupun digital. Ada dua jenis utama *rotary encoder*, yaitu : absolut dan *incremental* (relatif).

Sebuah *rotary encoder incremental* menyediakan output siklus (hanya) ketika *encoder* diputar. *Rotary encoder* dapat berupa mekanik atau optik. Jenis mekanik membutuhkan *debouncing* dan biasanya digunakan sebagai potensiometer digital pada peralatan termasuk perangkat konsumen.

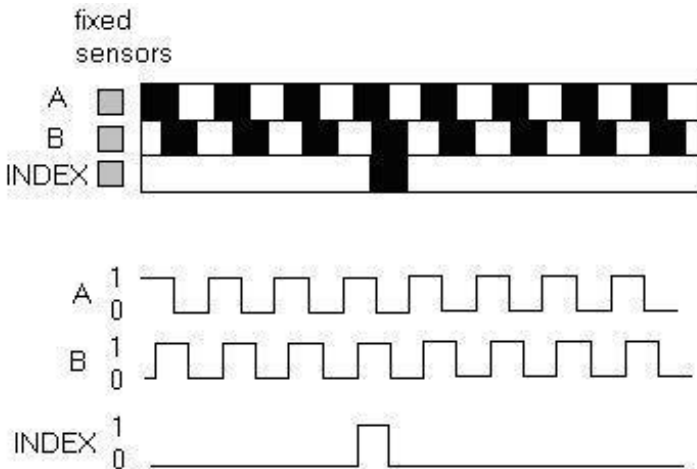
Rotary encoder incremental adalah yang paling banyak digunakan dari semua *rotary encoder* karena biaya rendah dan kemampuan untuk memberikan sinyal yang dapat dengan mudah ditafsirkan untuk memberikan informasi terkait seperti gerak kecepatan.

Encoders Incremental digunakan untuk melacak gerakan dan dapat digunakan untuk menentukan posisi dan kecepatan. Hal ini dapat berupa linear atau gerakan berputar. Karena arah dapat ditentukan, pengukuran yang sangat akurat dapat dibuat.

Rotary Encoder Incremental memiliki dua buah *output* yang disebut *channel A & channel B*, yang disebut *output quadrature*, karena pada *channel A* dan *channel B* memiliki perbedaan sebesar 90 derajat keluar dari fase.

Tabel 2.1. Diagram State Encoders Incremental

Coding untuk rotasi searah jarum jam			Coding untuk rotasi berlawanan arah jarum jam		
Tahap	A	B	Tahap	A	B
1	0	0	1	1	0
2	0	1	2	1	1
3	1	1	3	0	1
4	1	0	4	0	0



Gambar 2.3. Contoh Pola Keluaran *Incremental Encoder*

2.4 GSM Shield (Modem GSM SIM 900A) [5]

SIM 900A GSM/GPRS *Shield* adalah modul pengiriman data yang menggunakan jaringan telepon seluler GSM untuk menerima data dari lokasi yang jauh. Media data yang dapat dikirim melalui SIM 900A ini ada tiga media, yaitu :

- *Short Message Service*
- Audio
- Layanan GPRS



Gambar 2.4. Modem GSM SIM 900A

Pada SIM 900A GSM/GPRS telah dilengkapi oleh *simcard* layaknya *handphone*. Pada modem ini memerlukan tegangan sebesar 4,5 Volt.

SIM 900A GSM/GPRS *Shield* dikonfigurasi dan dikontrol melalui UART menggunakan perintah AT. Berdasarkan modul SIM900 dari SIMCOM, GPRS *Shield* bekerja seperti ponsel. Selain fitur komunikasi, GPRS *Shield* memiliki 12 GPIOs, 2 PWMs dan ADC.

Berikut ini adalah fungsi perintah AT yang berkaitan dengan proyek tugas akhir :

Tabel 2.2. Fungsi perintah AT pada SIM 900A

AT	Berfungsi sebagai pemberi perintah pada SIM 900A
ATE0	Berfungsi untuk mengaktifkan ECHO Mode. Yang mana maksud dari ECHO mode adalah apabila SIM 900A reset, maka konfigurasi

	akan berubah ke konfigurasi terakhir yang sebelumnya telah dimuat
AT + CMGF	Berfungsi untuk menentukan format SMS (<i>Short Message Service</i>). Format yang dapat dipergunakan adalah format PDU dan Text.
AT + CNMI	Berfungsi sebagai indikator apabila terdapat SMS (<i>Short Message Service</i>) baru.
AT + CMGS	Berfungsi untuk mengirim SMS (<i>Short Message Service</i>).

2.5 GearBox [6][7]

Gearbox berfungsi untuk menyalurkan tenaga atau daya mesin ke salah satu bagian mesin lainnya, sehingga unit tersebut dapat menghasilkan gerakan baik berupa putaran maupun pergeseran.

GearBox merupakan suatu alat khusus yang diperlukan untuk menyesuaikan daya atau torsi (momen/daya) dari motor yang berputar. Dapat pula dikatakan bahwa *gearbox* adalah alat pengubah daya dari motor yang berputar menjadi tenaga yang lebih besar.

Prinsip kerja *gearbox* adalah sebagai berikut : Putaran dari motor diteruskan ke *input shaft* (poros *input*) melalui hubungan antara *clutch* / kopling, kemudian putaran diteruskan ke *main shaft* (poros utama), torsi / momen yang ada di *main shaft* diteruskan ke spindel mesin, karena adanya perbedaan rasio dan bentuk dari gigi - gigi tersebut sehingga rpm atau putaran spindel yang dikeluarkan berbeda, tergantung dari rpm yang diinginkan.



Gambar 2.5. DC Planet Geared Motor Type PG 36

2.6 DC Motor Driver [8][9]

Motor DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada motor DC sedangkan arus keluaran pada mikro sangat kecil. Jika motor langsung dihubungkan ke mikrokontroler, maka hal ini akan menyebabkan kerusakan pada mikrokontroler tersebut. Motor *Driver* merupakan pilihan alternatif yang harus digunakan untuk mengendalikan motor DC.

Motor *Driver* merupakan suatu rangkaian khusus yang memiliki fungsi untuk mengatur arah ataupun kecepatan pada motor DC. Bentuk rangkaian motor *driver* yang umum digunakan yaitu H-Bridge. Berbentuk seperti huruf H yang memiliki perbedaan fungsi di setiap sisinya.

Motor *Driver* yang dipergunakan pada Tugas Akhir ini adalah *type* BTS7971B BTN7971B *High Power Motor Driver Module with High Speed Optical Coupling Isolation*.



Gambar 2.6. DC Motor Driver

Spesifikasi driver motor di atas adalah :

1. *Double* BTS/BTN7971 arus besar jembatan H driver;
2. *Highspeed* A2630 kopling optik, isolasi kopling optik dengan chip mikro tunggal, secara efektif melindungi mikrokontroler;
3. 78M05 *voltage regulator* chip, dengan daya kerja tambahan cahaya;
4. Daya Baterai dengan terminal *input*;
5. Driver motor *output* terminal (GND, OUT1 and OUT2);
6. Chip mikro tunggal untuk modul driver dengan 3 kabel (VIN, IN1 and IN2);
7. PCB dengan ukuran : 55.9 (mm) x 31.5 (mm)

2.7 LCD 8 x 2 [10]

LCD (*Liquid Crystal Display*) adalah modul yang dapat menampilkan visual secara elektronik melalui cairan *liquid* yang terdapat pada modul tersebut.



Gambar 2.7. LCD 8x2

LCD 8x2 adalah LCD yang terdiri dari 8 karakter dengan 2 baris sebagai *display* nya. Yang mana LCD ini memunculkan teks berwarna hitam dengan latar belakang (*background*) LCD adalah hijau. LCD 8 x 2 memiliki dimensi sebesar 1.574" x 1.411".

2.8 Reed Switch [11][12]

Reed Switch adalah saklar listrik yang dioperasikan oleh medan magnet. *Reed switch* terdiri dari dua kawat feromagnetik nikel - besi dan pisau kontak berbentuk khusus (buluh) diposisikan dalam kapsul kaca tertutup rapat.



Gambar 2.8. Reed Switch

Reed Switch dapat dioperasikan dengan menggunakan medan magnet yang dihasilkan oleh salah satu magnet permanen atau arus pembawa *coil*. *Reed Switch* berfungsi sebagai saklar yang aktif atau terhubung apabila di area jangkauannya terdapat medan

magnet. Medan magnet yang cukup kuat jika melalui area sekitar *reed switch*, maka dua buah plat yang saling berdekatan, akan terhubung sehingga akan memberikan rangkaian tertutup bagi rangkaian yang dipasangkannya.

2.9 *Limit Switch* [13]

Limit Switch adalah salah satu jenis sensor yang menggunakan sentuhan fisik agar dapat mengubah posisi saklarnya. Pada saat tuas (biasa disebut *cam*) mengenai atau menekan bagian kepala dari *limit switch*, maka sensor ini akan langsung bekerja sehingga kontak - kontak yang ada pada bagian dalamnya akan ikut pula bekerja. Pada saat sensor bekerja, dapat langsung dihubungkan ke perangkat atau komponen lainnya, seperti *solenoid valve* atau lampu indikator.



Gambar 2.9. *Limit Switch*

Limit switch memiliki tiga buah terminal, yaitu : *central terminal*, *normally close (NC) terminal*, dan *normally open (NO) terminal*. Sesuai dengan namanya, *limit switch* digunakan untuk membatasi kerja dari suatu alat yang sedang beroperasi. Terminal NC, NO, dan *central* dapat digunakan untuk memutuskan aliran listrik pada suatu rangkaian atau sebaliknya.

-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

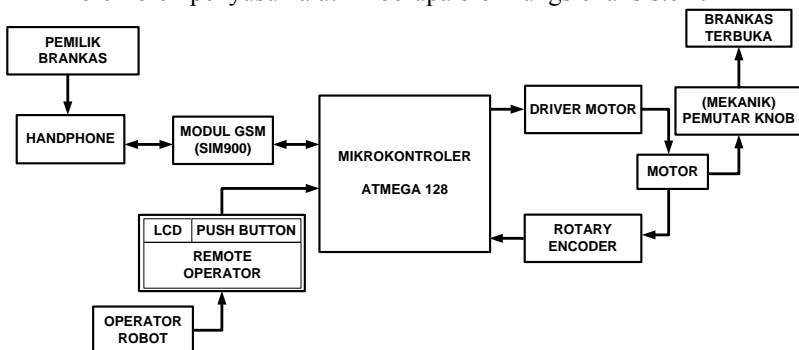
Perancangan alat pada bab ini terdiri dari perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

Perancangan perangkat keras (*hardware*) dibagi menjadi 2 bagian, yaitu meliputi perancangan elektronik dan juga perancangan mekanik. Perancangan elektronik terdiri dari skematik I/O mikrokontroler ATmega 128, pengkoneksian Modul GSM SIM 900A, pengendalian *planetary DC geared* motor menggunakan driver motor, pengkoneksian *voltage regulator*, serta perancangan sensor *limit switch* dan *reed switch*. Sedangkan untuk perancangan mekanik, meliputi perancangan pemegang *knob*, perancangan *casing* elektronik, dan perancangan bagian inti robot yang terdiri dari perancangan dudukan motor, perancangan pegangan robot untuk operator, dan juga perancangan penahan robot pada brankas.

Untuk perancangan perangkat lunak (*software*) meliputi perancangan pemrograman mikrokontroler ATmega 128, perancangan pemrograman modul GSM SIM 900A, Perancangan pemrograman *planetary DC geared* motor PG36 serta pemrograman tampilan pada LCD.

3.1 Blok Fungsional Sistem

Pada bab ini dibahas mengenai perancangan alat secara keseluruhan. Dijelaskan pula lebih terinci tiap-tiap bagian atau blok-blok penyusun alat ini berupa blok fungsional sistem.



Gambar 3.1. Blok Fungsional

Gambar 3.1 merupakan blok fungsional untuk perancangan robot pembuka brankas. Diagram blok tersebut memiliki tahapan sebagai berikut :

1. Pemilik brankas mengirim SMS (*Short Messages Service*) berupa kode putar brankas melalui *handphone* nya. Dimana kode putar terdiri dari beberapa digit angka yang sesuai dengan kode putar brankas tersebut.
2. SMS (*Short Messages Service*) tersebut dikirimkan ke nomor yang digunakan oleh Modul GSM SIM 900A.
3. Proyek Tugas Akhir ini terdiri bagian elektronik dan bagian mekaniknya. Dimana bagian elektroniknya ini terdiri beberapa *push button* yang dihubungkan ke SIM 900A, *driver* motor, dan juga mikrokontroler ATmega 128. Selain itu bagian elektroniknya juga terdapat LCD yang akan menampilkan indikator kerja robot. Bagian elektronik ini disusun dan diletakkan di sebuah kotak yang terbuat dari *acrylic*.
4. Setelah SMS diterima oleh robot, maka DC Planet *Geared* Motor akan berputar sesuai dengan kode putar yang dikirim oleh pemilik brankas.
5. Pada bagian mekaniknya, terdapat pemutar *knob*, penahan robot pada pintu brankas menggunakan dua buah penyedot/penghisap dengan modifikasi *holder*. Selain itu juga terdapat kaki penahan robot yang mana akan membantu agar robot semakin kuat tertahan pada pintu brankas.
6. Dan pada tahapan terakhirnya, sinkronisasi bagian elektronik dan bagian mekaniknya menjadikan brankas terbuka. Dengan kondisi robot tetap terpasang pada pintu brankas.

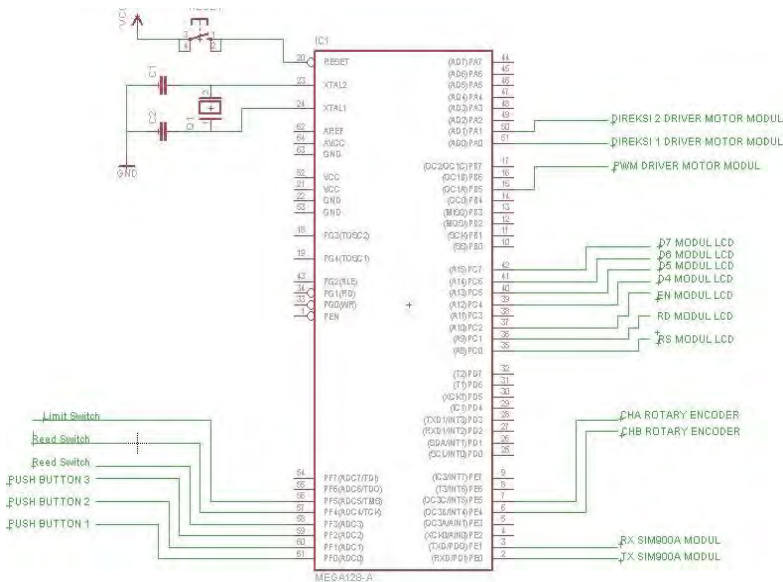
3.2 Perancangan Perangkat Keras (*Hardware*)

Perancangan Perangkat Keras (*Hardware*) pada Tugas Akhir ini terdiri dari perancangan elektronik dan perancangan mekanik. Perancangan elektronik meliputi skematik I/O Mikrokontroler ATmega 128, pengkoneksian Modul GSM SIM 900A, pengendalian motor menggunakan modul *driver* motor, perancangan *voltage regulator*, serta perancangan sensor *limit switch* dan *reed switch*. Selanjutnya pada perancangan mekanik meliputi perancangan pemegang *knob*, perancangan *casing* elektronik, dan perancangan bagian inti robot yang mana perancangan ini terdiri dari perancangan dudukan motor,

perancangan pegangan robot untuk operator dan perancangan penahan robot pada brankas.

3.2.1 Perancangan Elektronik

3.2.1.1 Skematik Mikrokontroler ATmega 128



Gambar 3.2. Skematik Mikrokontroler ATmega 128

Pada Gambar 3.2 Skematik Mikrokontroler ATMEGA 128 dapat diketahui mengenai penyambungan / pengkoneksian *port - port* yang digunakan pada LCD, 4 buah *push button*, *driver motor*, *rotary encoder* dan juga pada modul GSM SIM 900A. Pengkoneksian awal adalah dengan VCC 5V mikrokontroler sebagai sumber.

Pin - pin pada mikrokontroler ATmega 128 yang digunakan pada Tugas Akhir ini adalah sebagai berikut :

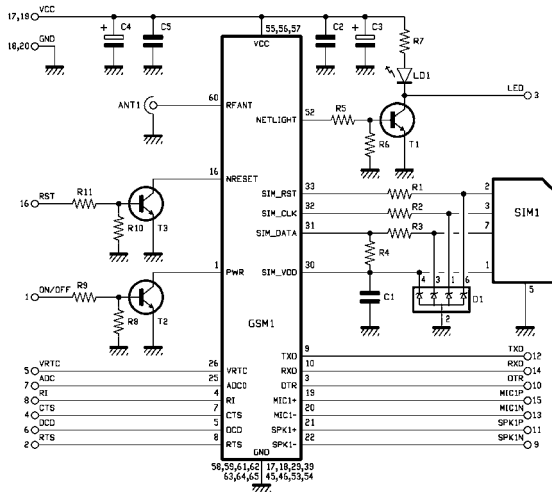
Tabel 3.1. Pin Mikrokontroler ATmega 128 yang digunakan

PIN		FUNGSI
2	PE0 (RXD/PD1)	Tx Modul SIM 900A
3	PE1 (TXD/PD0)	Rx Modul SIM 900A
6	PE4 (DC3B/INT4)	Channel B <i>Rotary Encoder</i>
7	PE5 (DC3C/INT5)	Channel A <i>Rotary Encoder</i>
15	PB5 (OC1A)	PWM <i>Driver</i> Motor Modul
20	Reset	Tombol Reset
23	XTAL2	<i>Ground</i>
24	XTAL1	<i>Ground</i>
35	PC0 A8	RS Modul LCD
36	PC1 A9	RD Modul LCD
37	PC2 A10	EN Modul LCD
39	PC4 A12	D4 Modul LCD
40	PC5 A13	D5 Modul LCD
41	PC6 A14	D6 Modul LCD
42	PC7 A15	D7 Modul LCD
50	PA1 AD1	Direksi 2 <i>Driver</i> Motor Modul
51	PA0 AD0	Direksi 1 <i>Driver</i> Motor Modul
56	PF5 (ADC5/TMS)	Sensor <i>Limit Switch</i>
57	PF4 (ADC4/TCK)	Sensor <i>Reed Switch</i>
58	PF3 ADC3	Sensor <i>Reed Switch</i>
59	PF2 ADC2	<i>Push Button</i> 3
60	PF1 ADC1	<i>Push Button</i> 2
61	PF0 ADC0	<i>Push Button</i> 1

3.2.1.2 Pengkoneksian Modul GSM SIM 900A

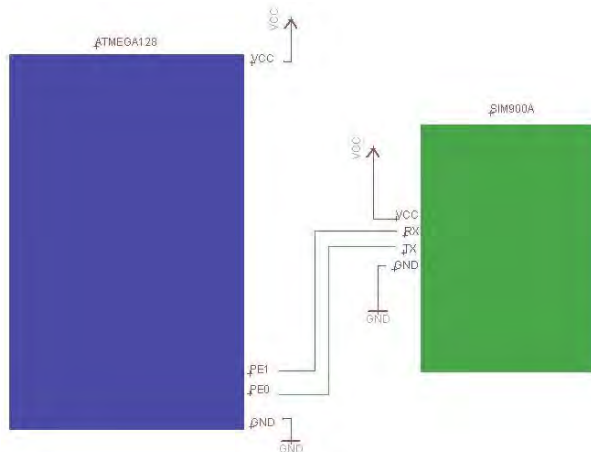
Yang dimaksud dengan pengkoneksian modul GSM SIM 900A adalah mengkoneksikan modul SIM900A dengan mikrokontroler

ATMega 128. Berikut ini adalah skematik Modul GSM SIM 900A :



Gambar 3.3. Skematik Modul GSM SIM 900A

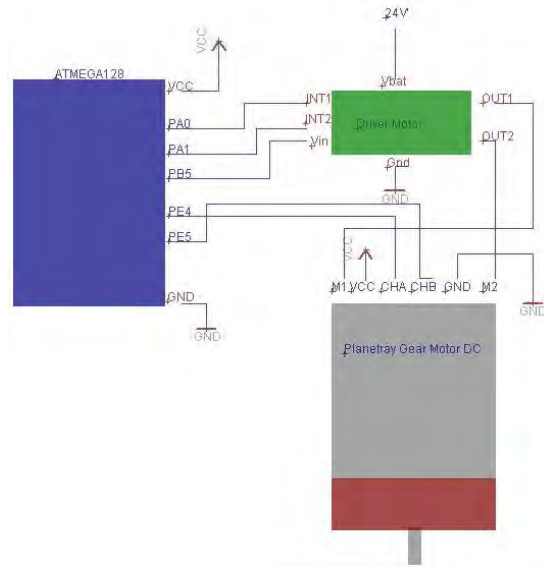
Dan berikut adalah *wiring* pengkoneksian modul GSM SIM 900A dengan kontroler :



Gambar 3.4. Wiring Pengkoneksian SIM 900A dengan kontroler

3.2.1.3 Pengendalian DC Planet *Geared* Motor PG36 Menggunakan Modul *Driver* Motor

Driver motor berfungsi sebagai pengontrol arah putaran dan kecepatan motor yang mana pada tugas akhir ini motor yang dipergunakan adalah DC Planet *Geared* Motor PG36. Berikut adalah konfigurasi dari *Driver* Motor :



Gambar 3.5. Wiring konfigurasi *Driver* Motor

3.2.1.4 Pengkoneksian *Voltage Regulator*

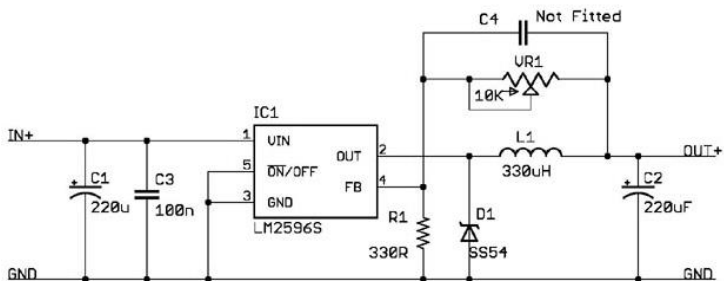
Pengatur tegangan (*Voltage Regulator*) berfungsi menyediakan suatu tegangan keluaran DC yang tetap yang tidak dipengaruhi oleh perubahan tegangan masukan, arus beban keluaran, dan suhu. Pengatur tegangan adalah salah satu bagian dari rangkaian catu daya DC.

Berikut ini adalah IC yang dipergunakan untuk merancang *Voltage Regulator*. IC LM2596.



Gambar 3.6. Voltage Regulator IC LM2596

Skematik *Voltage Regulator* IC LM2596 adalah sebagai berikut :



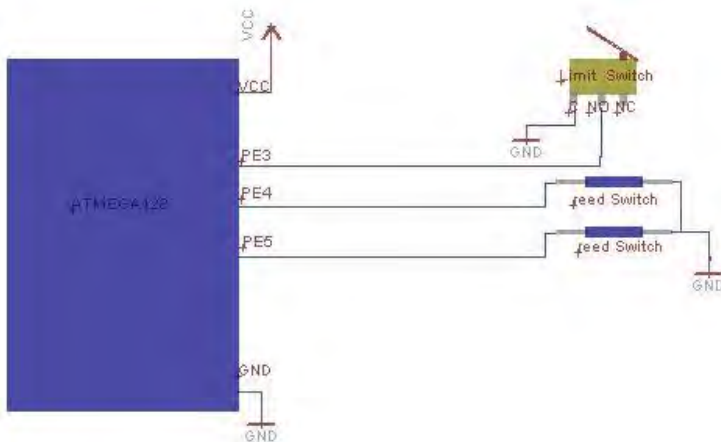
Gambar 3.7. Skematik Voltage Regulator IC LM2596

Pada rancangan ini inputan dari baterai lipo sebesar 22,2 Volt dan kemudian setelah melalui *voltage regulator* maka dihasilkan tegangan sebesar 12 V.

3.1.2.5 Perancangan Sensor *Limit Switch* dan *Reed Switch*

Sensor *limit switch* ini berfungsi untuk menjaga agar robot pembuka brankas ini tepat terpasang pada pintu brankas. Dimana sensor *limit switch* ini diletakkan pada papan aluminium diantara kaki robot.

Sedangkan sensor *reed switch* dipasang pada penutup *security* kode putar. Sensor *reed switch* ini yang menjadi indikasi apabila terjadi kecurangan dari operator saat membuka robot. Berikut ini adalah *wiring* sensor *limit switch* dan juga sensor *reed switch*.

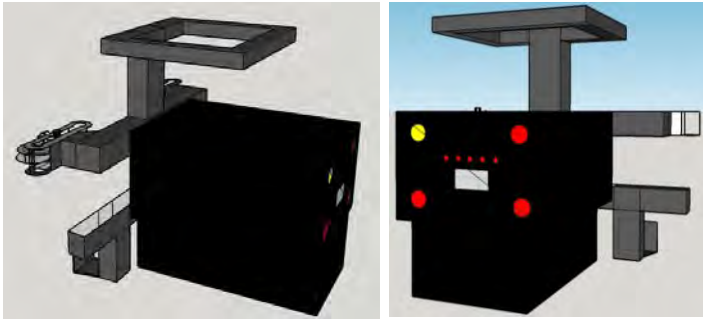


Gambar 3.8. Wiring Sensor *Limit Switch* dan *Reed Switch*

3.2.2 Perancangan Mekanik

Perancangan mekanik pada Robot Pembuka Brankas ini terdiri dari perancangan pemegang *knob*, perancangan *casing* elektronik, dan perancangan bagian inti robot. Dimana pada bagian perancangan bagian inti robot terdiri dari perancangan dudukan motor, perancangan pegangan robot untuk operator, perancangan penahan robot pada brankas, dan perancangan kaki penyangga robot.

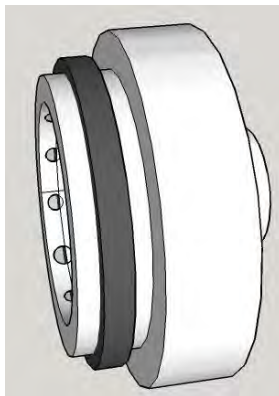
Berikut ini adalah gambaran alat robot pembuka brankas tersebut :



Gambar 3.9. Robot Pembuka Brankas

3.2.2.1 Perancangan Pemegang *Knob*

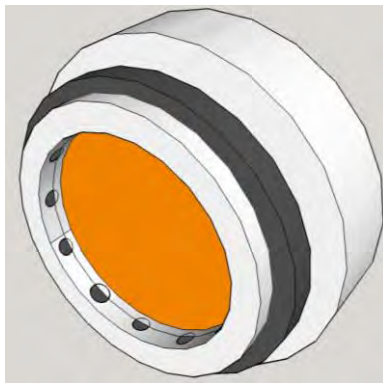
Perancangan pemegang *knob* berfungsi untuk memegang *knob* dari brankas yang mana pemegang *knob* ini sebagai pencekram agar putaran dari DC Planet *Geared* Motor dapat berputar sesuai dengan kode yang diterima oleh Modul GSM SIM 900A. Yang dimaksud dengan *knob* adalah suatu pegangan atau tombol yang terdapat pada brankas. *Knob* pada brankas dapat diputar ke kanan dan ke kiri yang berisikan angka - angka yang berfungsi sebagai password. Berikut ini adalah desain dari pemegang *knob* tersebut :



Gambar 3.10. Pemegang *Knob*

Dari gambar 3.10 Pemegang *knob* di atas, dijelaskan bahwa perancangan dan pembuatan pemegang *knob* adalah sebagai berikut :

1. Pemegang *knob* terbuat dari 2 buah nilon berwarna putih susu dengan nilon pertama memiliki diameter luar sebesar 12 cm dan diameter dalam sebesar 10,3 cm. Memiliki tebal 17 mm. Nilon kedua berdiameter lebih kecil dengan diameter luar sebesar 10,3 cm. Dan diameter bagian dalamnya sebesar 8 cm. Dengan tebal 12 mm.
2. Pada nilon kedua (yang berukuran lebih kecil) sebagai pemegang *knob*, dilubangi sebanyak 16 lubang. Masing - masing lubang tersebut berdiameter 5 mm dengan panjang sebesar tebal dari nilon tersebut (12 mm)
3. Lubang bagian dalam nilon tersebut diberi per dengan panjang 10 mm dan kemudian ditutupi dengan peluru yang dikaitkan menggunakan *ring* dari aluminium agar per menjadi merenggang dan peluru dapat tertanam dengan kokoh.
4. Tujuan memberi per dan peluru pada lubang nilon agar *knob* menjadi tidak licin dan pemegang *knob* dapat mencengkram dengan kuat.



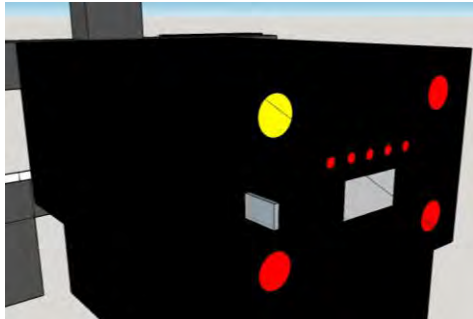
Gambar 3.11. Peluru pada pemegang *knob*

3.2.2.2 Perancangan *Casing Elektronik*

Perancangan *casing* elektronik berfungsi sebagai tempat/wadah komponen - komponen elektronik yang dipergunakan. Perancangan *casing* ini meliputi perancangan *box* sebagai wadah bagian elektronik, perancangan *box* sebagai pelindung *DC Planet Geared Motor*, dan juga penyangga/dudukan untuk bagian elektronik itu sendiri.

Proses perancangan dan pembuatan *casing* elektronik adalah sebagai berikut :

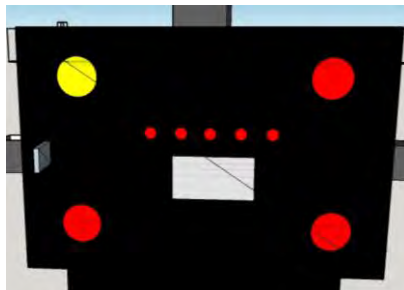
1. Perancangan *box* sebagai wadah elektronik terbuat dari *acrylic* berwarna hitam dengan tebal *acrylic* 2 mm berukuran 15 x 16 x 7 cm. *Box acrylic* ini didesain sedemikian rupa sehingga memudahkan operator untuk memeriksa apabila terjadi *error* pada robot tersebut.



Gambar 3.12. *Box Acrylic Elektronik*

2. Pada *box acrylic* terdapat 4 buah *push button* yang mana *push button* ini berfungsi untuk mengoperasikan robot tersebut. *Push button* berwarna kuning sebagai tombol reset robot. Dan *push button* merah di bawah *push button* kuning berfungsi sebagai tombol start untuk mulai menjalankan motor. Selanjutnya *push button* bagian kanan (*push button A* dan *push*

button B) yang dapat dipergunakan sebagai tombol opsional. Kemudian di sisi depan juga terdapat *mini voltage detector* yang mana berfungsi untuk menampilkan voltase yang tersisa pada baterai lipo. Terdapat pula *push button* utama yang berfungsi sebagai tombol on - off sistem. Jadi ketika *push button* diaktifkan maka baterai akan aktif sehingga mengaktifkan sistem elektronik pada robot. Selain itu juga terdapat LED berwarna merah berjumlah 5 buah yang mana LED ini adalah indikator sistem. Dimana LED nomor 3 (bagian tengah) akan menyala apabila sistem menyala. Kemudian LED nomor 1,2,4 dan 5 akan menyala bergantian sebagai indikator kerja sistem. Setelah semua LED menyala, LED kemudian akan mati dan sistem sudah berjalan. Selanjutnya LED akan menyala secara berurutan mulai LED 1 sampai LED ke 5 yang menjadikan indikator *step* putaran motor untuk membuka brankas. Berikut ini adalah bagian depan *box acrylic* elektronik :

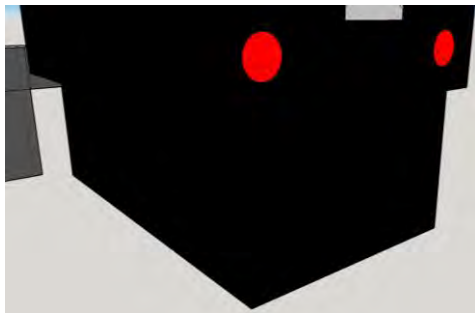


Gambar 3.13. Box Acrylic sisi depan

3. *Box acrylic* ini dikaitkan dengan penyangga/ dudukan elektronik yang terbuat dari lembaran aluminium dengan ukuran sebesar 10 cm dan tebal 3 mm dengan bentuk persegi panjang. Namun penyangga yang terbuat dari

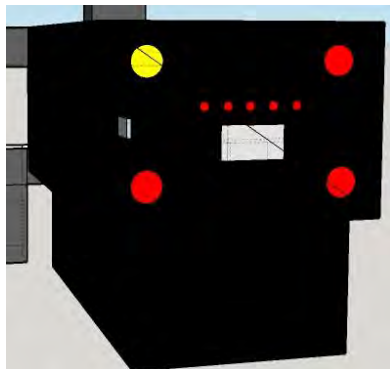
aluminium ini tidak terlihat dari luar. Karena sudah tertutupi dengan box *acrylic* pelindung *DC Planet Geared Motor*.

4. Selanjutnya terdapat pula box *acrylic* sebagai pelindung *DC Planet Geared Motor* yang ukurannya lebih kecil daripada box *acrylic* elektroniknya. Dengan ukuran 30x9,3x7 cm menggunakan *acrylic* berwarna hitam dengan tebal 2 mm.



Gambar 3.14. Box Acrylic pelindung motor

5. Box *acrylic* pelindung motor terletak di bawah box *acrylic* elektronik. Yang mana pemasangannya menggunakan mur dan baut agar lebih kuat dan kokoh.



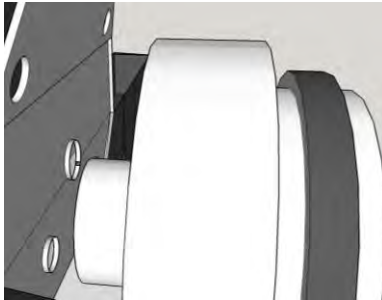
Gambar 3.15. Box Acrylic keseluruhan

3.2.2.3 Perancangan Bagian Inti Robot

Perancangan bagian inti robot terdiri dari :

1. Perancangan Dudukan Motor

Perancangan dudukan motor yang dimaksudkan di sini adalah pengait motor pada bagian inti robot dan penyatu dengan nilon pegangan *knob*.



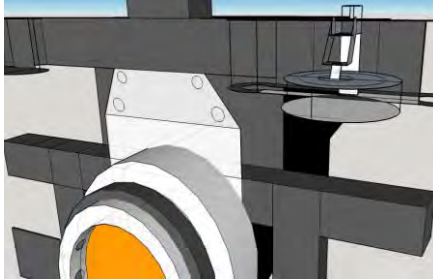
Gambar 3.16. Perancangan Dudukan Motor

Dudukan motor ini terbuat dari aluminium dengan tebal 2 mm. Berbentuk trapesium sama sisi dengan ukuran 10 cm x 5 cm yang direkatkan dengan bagian inti robot menggunakan mur dan baut berjumlah 3 buah. Selanjutnya bagian tersebut dikaitkan pula dengan nilon kecil berbentuk lingkaran dengan diameter 3 cm dan panjang 1,7 cm.

Dikaitkan dengan nilon kecil dimaksudkan sebagai penghubung antara *DC Planet Geared Motor* dengan pemegang *knob*.

Lempengan aluminium sebagai dudukan motor ini dihubungkan dengan batangan aluminium vertikal dengan tebal 1 cm dan panjang 24 cm. Dimana batangan aluminium ini yang nantinya juga menjadi penyangga penghubung dengan kaki robot.

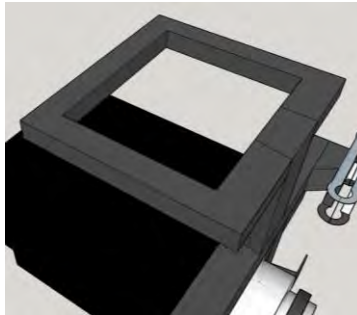
Berikut ini adalah gambar perancangan dari batangan aluminium tersebut :



Gambar 3.17. Batangan Aluminium Penyangga

2. Perancangan Pegangan Robot untuk Operator
 Perancangan pegangan robot untuk operator yang dimaksudkan adalah bagian yang dapat digunakan oleh operator sebagai pegangan robot agar dapat dibawa dengan mudah.

Desain dari pegangan robot ini adalah sebagai berikut :



Gambar 3.18. Pegangan Robot untuk Operator

Dari gambar di atas, dapat diketahui bahwa :

1. Pegangan tersebut terbuat dari batangan aluminium dengan tebal 2 mm. Batangan aluminium tersebut dibentuk persegi panjang dengan ukuran 12,5 x 7,7 x 2,5 cm.

2. Pembentukan pegangan ini masing - masing sikunya dikaitkan dengan lempengan aluminium yang berbentuk siku - siku menggunakan *blind rivet* dengan ukuran 3,2 mm x 9,5 mm. Dimana setiap siku batangan aluminium tersebut diberi 4 buah *blind rivet*.
3. Kemudian pegangan ini dikaitkan pada batangan aluminium horizontal inti robot dengan menggunakan lempengan aluminium yang berbentuk siku - siku menggunakan *blind rivet* pula. Dengan jumlah *blind rivet* sebanyak 4 buah. Sehingga pegangan robot ini menyatu dengan bagian robot itu sendiri.

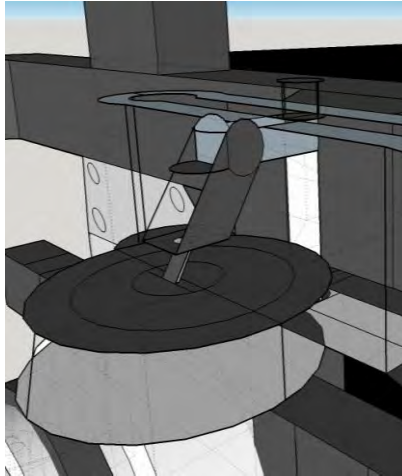
3. Perancangan Penahan Robot pada Brankas

Perancangan penahan brankas terdiri dari dua buah *holder* penghisap. Dimana *holder* penghisap ini diletakkan pada sisi kanan dan sisi kiri robot selayaknya lengan robot.

Holder penghisap ini terbuat dari modifikasi dua buah holder *handphone*. Yang mana penghisapnya itu sendiri berbentuk bulat dengan diameter sebesar 7,3 cm. Sedangkan *holder* nya sendiri memiliki panjang lengan 8 cm.



Gambar 3.19. Penghisap pada Lengan Robot

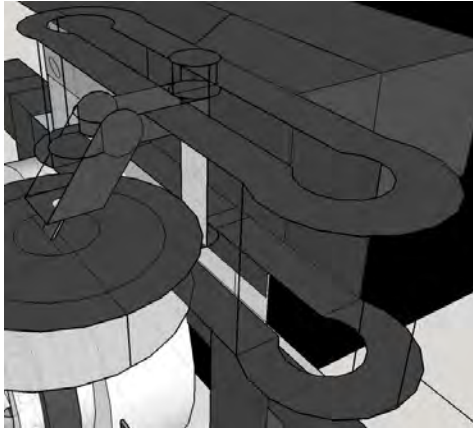


Gambar 3.20. *Holder* pada Lengan Robot

Cara pemasangan *holder* penghisap pada robot menggunakan mur pengunci yang terpasang pada *holder* penghisap tersebut. Yang mana *holder* ini dapat digerakkan secara horizontal dan vertikal.

Pada penahan robot ini, terdapat sebuah *space* yang berfungsi agar *holder* penghisap dapat diatur bergeser ke kanan atau ke kiri atau dengan kata lain jangkauan lengan penahan menjadi panjang atau pendek.

Space pada lengan robot ini terbuat dari acylic dengan tebal 12 mm dan dengan ukuran 14 x 12 cm berbentuk *elips* dimana bagian tengahnya terdapat ruang kosong dengan panjang 13 cm dan lebar 0,5 cm yang berfungsi agar mur dari *holder* penghisap ini dapat digerakkan sesuai dengan keinginan. Desain bagian penggerak lengan robot dapat dilihat pada gambar berikut :

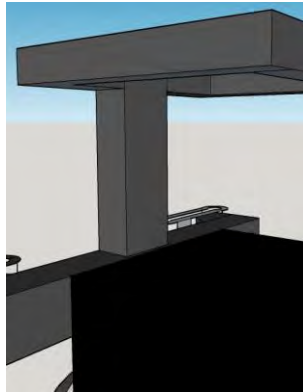


Gambar 3.21. Space Penggerak Lengan Robot

Penyambungan antara lengan robot dan bagian inti robot adalah dengan menggunakan sebuah batangan aluminium dengan tebal 12 mm dengan panjang 24,5 cm. Batangan aluminium yang berfungsi sebagai pengait ini diletakkan melintang pada bagian inti robot dan dikaitkan menggunakan lempengan aluminium dengan bentuk trapesium siku - siku di sisi kanan dan kiri. Dimana lempengan tersebut dipasang dibagian atas dan bawah. Lempengan aluminium tersebut memiliki tebal 1 mm dan dikaitkan dengan mur baut dengan masing - masing lempengan terdapat 8 mur baut.

Selain di sisi kanan dan kirinya, di bagian tengah batangan aluminium tersebut juga dikaitkan dengan lempengan aluminium agar lebih kuat lagi. Bentuk dari lempengan aluminium ini adalah trapesium sama kaki. Dengan cara penyambungan dengan menggunakan *blind rivet* dengan ukuran 3,2 mm x 9,5 mm. Lempengan tersebut diletakkan dibagian atas dan bawah.

4. Setelah pada bagian penahan robot pada brankas dihubungkan dengan batang aluminium yang melintang. Yang mana bagian ini pula disambungkan dengan batang aluminium yang diletakkan tegak. Sehingga bagian ini menyatu dengan *box* elektronik. Batang aluminium yang diletakkan tegak ini memiliki ukuran sebesar 16,5 cm dengan tebal aluminium adalah 1 cm.



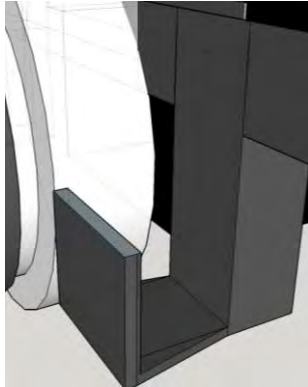
Gambar 3.22. Batang Aluminium Penghubung

4. Perancangan Kaki Penyangga Robot

Selain terdapat lengan pada robot yang berfungsi untuk menahan robot pada brankas, robot pembuka brankas ini juga dilengkapi dengan kaki penyangga robot. Yang mana kaki penyangga ini berfungsi sebagai penyangga robot dan juga berfungsi sebagai penahan robot pada brankas. Sehingga robot pembuka brankas ini memiliki dua penahan pada brankas. Lengan robot dan juga kaki robot.

Kaki pada robot ini berjumlah 2 buah yang diletakkan di bagian kanan dan kiri tubuh robot tersebut. Kaki penyangga ini terbuat dari batang aluminium yang dibentuk siku - siku. Yang mana bagian horizontal dengan panjang 11

cm dan tebal 1,3 cm. Serta pada bagian vertikalnya 7,5 cm dengan tebal yang sama dengan bagian horizontalnya. Berikut adalah design dari kaki robot tersebut :



Gambar 3.23. Kaki Penyangga Robot

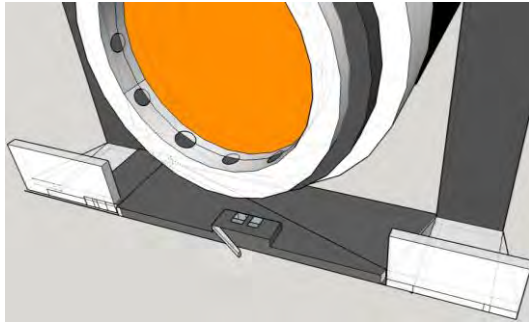
Cara penyambungan batangan siku - siku ini menggunakan lempengan aluminium berbentuk trapesium siku - siku yang dikaitkan dengan menggunakan *blind rivet* dengan ukuran 3,2 mm x 9,5 mm berjumlah 4 buah.

Selanjutnya pada bagian vertikal batangan aluminium dikaitkan pula dengan aluminium siku - siku 3 mm yang dibentuk trapesium sama kaki.

Dibagian kaki robot ini diletakkan pula *acrylic* berwarna hitam yang terdapat sensor *reed switch*. Yang mana apabila *acrylic* ini berfungsi untuk menutupi *knob* apabila sedang berputar. Sehingga operatr tidak dapat mengetahui kode putar dari brankas tersebut.

Selain itu di antara kaki robot bagian kanan dan kiri disatukan dengan pemasangan aluminium dengan ukuran 14,4 x 8,5 cm. Dimana pada aluminium ini terdapat sensor *limit switch*. Sensor *limit switch* aktif apabila robot ini sudah benar - benar terpasang dengan tepat pada

brankas yang mana sensor akan tertekan sehingga menjadi aktif.



Gambar 3.24. Alas Aluminium Sensor *Limit Switch*

3.3 Perancangan Perangkat Lunak (*Software*)

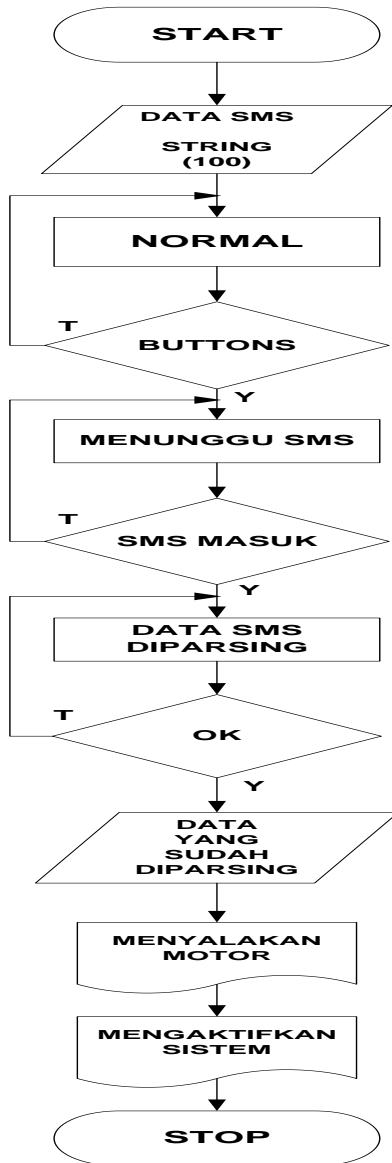
Perancangan perangkat lunak (*software*) pada Tugas Akhir ini terdiri dari pemrograman mikrokontroler ATmega 128, pemrograman DC Planet *Geared* Motor, Pemrograman Modul GSM SIM 900A, serta Pemrograman tampilan LCD.

3.3.1 Pemrograman Mikrokontroler ATmega 128

Perangkat lunak (*software*) yang digunakan dalam pemrograman Mikrokontroler ATmega 128 menggunakan *software Code Vision AVR* dan dihubungkan dari PC/laptop ke mikrokontroler ATmega 128 menggunakan *downloader*.

Pemrograman mikrokontroler ATmega 128 meliputi pemrograman *rotary encoder*, modul GSM SIM 900A, inputan berupa *push button*, tampilan LCD, dan juga DC Planet *Geared* Motor.

Berikut ini *flowchart* program keseluruhan:



Gambar 3.25. Flowchart program keseluruhan

3.3.2 Pemograman DC Planet *Geared* Motor

Pemograman DC Planet *Geared* Motor juga diprogram pada *Code Vision AVR*. Pemograman ini dilakukan dengan tujuan untuk mengatur putaran motor CW atau CCW. Selain itu mengatur pula kecepatan motor serta putaran motor agar sesuai dengan kode yang diterima oleh Modul GSM SIM 900A dan langsung diproses.

Berikut ini adalah contoh program yang dipergunakan untuk mengantur putaran motor CW atau CCW. Untuk program lainnya terlampir.

```
void jalan_motor(){
hitung_pulsa();
switch (index) {
case 1:
while (rotary[kiri] <= nilai1){ //nilai1=350
m_ccw();
PWM = 5;
ftoa(rotary[kiri],1,temp);lcd_gotoxy(0,0);lcd_puts(temp);
}
m_stop();
delay_ms(2000);
index = 2;
break;
case 2:
while (rotary[kiri] >= nilai2){ //nilai2=200
m_cw();
PWM = 5;
ftoa(rotary[kiri],1,temp);lcd_gotoxy(0,0);lcd_puts(temp);
}
m_stop();
delay_ms(2000);
index = 3;
break;
case 3:
lcd_clear();
break;
```

```

default:
};
}

```

3.3.3 Pemograman Modul GSM SIM 900A

Sama halnya dengan pemograman DC Planet *Geared* Motor, program Modul GSM SIM 900A ini dibuat di *Code Vision AVR*. Program dimaksudkan agar Modul GSM SIM 900A dapat membaca SMS yang diterima dengan baik. Kartu provider yang digunakan pada Modul GSM SIM 900A menggunakan kartu Indosat dengan nomor 085732020568. Dengan nomor *user* pemilik brankas adalah 085646450650.

Berikut ini adalah salah satu diantara beberapa program yang dipergunakan mengenai kontrol Modul GSM SIM 900A.

```

void kirim_sms(int command){
lcd_clear();lcd_gotoxy(0,0);lcd_putsf("Sending SMS");
switch (command) {
case 1:
printf ("AT+CMGS=");
putchar ("");printf ("+6285250077040");putchar

("");
putchar (13);delay_ms (1000);
printf ("Sistem telah diselesaikan dengan baik");
putchar (26);
//putchar(10);
putchar (13);
//while (getchar() != 'K'){
delay_ms(100);
break;
case 2:
printf ("AT+CMGS=");
putchar ("");printf ("+6285646450650");putchar

("");
putchar (13);delay_ms (1000);

```

```

printf("Data tidak diterima");
putchar (26);
//putchar(10);
putchar (13);
//while (getchar() != 'K'){
delay_ms(100);

break;
default:
};

}

```

Pada program di bawah ini adalah program mengenai penggunaan SIM 900A yang mana pada *case 1* berfungsi untuk konfirmasi kepada *user* bahwa sistem telah diselesaikan dengan baik dan brankas telah terbuka. Dan pada *case 2* adalah konfirmasi bahwa sistem tidak bekerja karena ada beberapa kenadala misalnya apabila penutup pelindung *knob* dibuka yang berarti sistem *security* tidak aman sehingga sistem tidak mengeksekusi

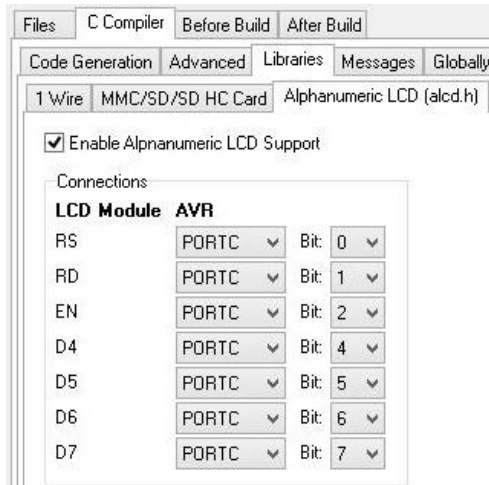
3.3.4 Pemograman Tampilan LCD

Salah satu jenis proyeksi yang dipergunakan untuk menampilkan data dari kontroler adalah LCD 8x2. Alasan mengapa memilih LCD 8x2 ini karena dimensi pada LCD ini sesuai dengan jumlah data yang ditampilkan dalam Proyek Tugas Akhir ini.

Pada awal perancangan perangkat lunak menggunakan *Code Vision AVR*, terlebih dahulu diperlukan konfigurasi untuk jenis mikrokontroler, beserta nilai *clock* yang digunakan, kemudian juga diperlukan konfigurasi *port* mikrokontroler dan konfigurasi untuk fitur – fitur yang akan digunakan seperti misalnya LCD.

Untuk konfigurasi tersebut, *Code Vision AVR* menyediakan fitur khusus untuk konfigurasi nilai – nilai ataupun modus apa yang diperlukan dalam pemrograman sehingga sangat memudahkan pengguna dalam

menggunakan fitur – fitur yang terdapat dalam AVR tersebut. Gambar 3.26 menunjukkan konfigurasi yang dilakukan menggunakan *CodeWizard AVR*.



Gambar 3.26. Konfigurasi Code Wizard AVR

Contoh program LCD yang dipergunakan adalah sebagai berikut :

```
lcd_clear();

lcd_gotoxy(0,0);lcd_puts("Starting");
```

BAB IV

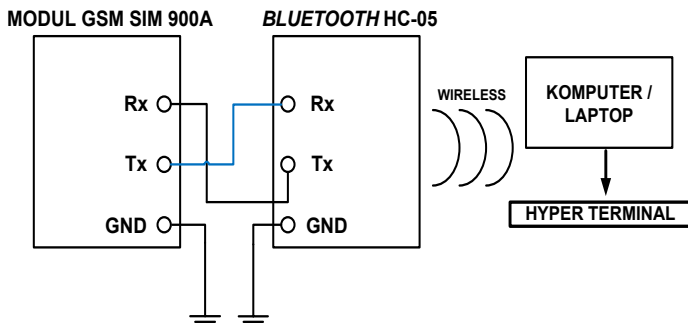
PENGUJIAN DAN ANALISA ALAT

Pada bab ini akan dibahas mengenai pengujian dan analisa alat dari robot pembuka brankas yang telah dibuat. Pengujian alat ini ditujukan untuk memastikan agar peralatan yang dibuat dapat berfungsi dengan baik.

Pengujian robot pembuka brankas ini meliputi pengujian modul GSM SIM 900A, pengujian DC planetary *geared* motor, pengujian *driver* motor, pengujian program *driver* motor terhadap DC planetary *geared* motor, pengujian program modul GSM SIM 900A terhadap kontroler, dan pengujian kepresisian putaran motor. Setelah melakukan beberapa pengujian alat, data yang diperoleh akan dianalisa untuk mengetahui proses kerja dari seluruh sistem alat yang dibuat.

4.1 Pengujian Modul GSM SIM 900A

Pengujian Modul GSM SIM 900A dilakukan melalui *hyperterminal* pada komputer/laptop. Cara pengujian Modul GSM SIM 900A yaitu dengan menggunakan *bluetooth* HC-05 dimana komunikasi yang digunakan adalah komunikasi serial dengan level tegangan TTL. Berikut ini adalah gambar *wiring* diagram pengkoneksian modul GSM SIM 900A.

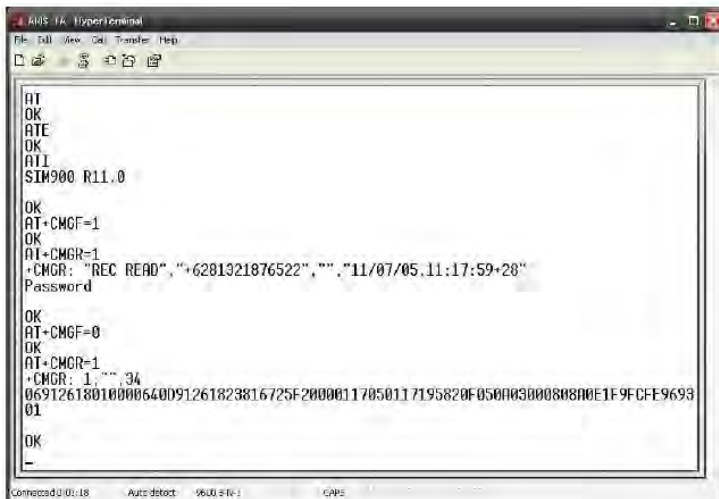


Gambar 4.1. Wiring Diagram Pengkoneksian

Cara pengujian Modul GSM SIM 900A dengan cara menghubungkan Vcc dari Modul GSM SIM 900A ke Vcc dari

bluetooth HC-05, *ground* dari Modul GSM SIM 900A ke *ground* dari *bluetooth* HC-05, Rx dari Modul GSM SIM 900A ke Tx dari *bluetooth* HC-05, dan Tx dari Modul GSM SIM 900A ke Rx dari *bluetooth* HC-05. Kemudian setelah terhubung, *bluetooth* dikoneksikan dengan laptop/komputer yang sudah diinstall dengan Aplikasi *Hyperterminal*.

Berikut adalah hasil pengujian SIM 900A dengan menggunakan *AT Command* pada komputer menggunakan *Hyperterminal*. Data-data yang ditampilkan pada program aplikasi *Hyperterminal* yang diterima oleh komputer dapat dilihat pada Gambar 4.2. sebagai berikut :



Gambar 4.2. Tampilan pada *Hyperterminal*

Keterangan :

1. AT adalah perintah untuk menguji koneksi modul GSM SIM 900 terhadap *software* di komputer. Respon "OK" menandakan bahwa koneksi dalam kondisi baik.
2. ATI adalah perintah untuk melihat tipe SIM yang digunakan.
3. AT+CMGF=1 adalah perintah untuk memerintahkan modul GSM SIM 900 bekerja pada SMS *mode* teks. Respon "OK" menandakan perintah berhasil di eksekusi. Jika respon

“*ERROR*” menandakan bahwa modul GSM belum terhubung dengan baik.

4. AT+CMGR adalah untuk membaca sms pada lokasi tertentu dari memori pesan. AT+CMGR=1 adalah perintah untuk membaca sms pada memori pesan *index* 1.
5. REC READ adalah *Received Read Message* (pesan diterima sudah dibaca). Sedangkan untuk “REC UNREAD” adalah *Received Unread Message* (pesan diterima sebelum dibaca).
6. AT+CMGF=0 adalah perintah untuk memerintahkan modem GSM bekerja pada SMS mode PDU (*Protokol Data Unit*). Respon “OK” menandakan perintah berhasil di eksekusi. Jika respon “*ERROR*” menandakan bahwa modem belum terhubung dengan baik.

4.2 Pengujian *Driver Motor*

Tujuan dari pengujian *driver* motor adalah untuk menguji *driver* motor dapat berkerja dengan baik atau tidak ketika digunakan. Pengujian *driver* motor dilakukan dengan cara mengukur perbedaan potensial antara output 1 dan output 2. Berikut ini adalah skematik *driver* motor :



Gambar 4.3. Modul *Driver Motor*

Hasil Pengujian *Driver Motor* adalah sebagai berikut :

Tabel 4.1. Hasil Pengujian *Driver Motor*

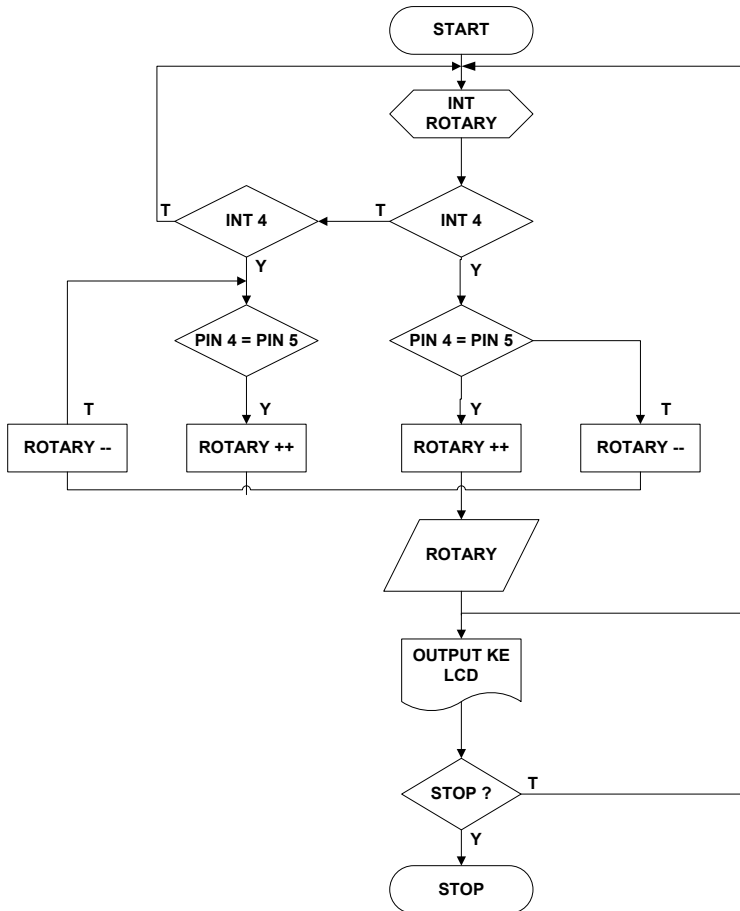
No.	INT 1	INT 2	Vin	Vout	Kondisi Motor
1.	0V	0V	0V	24,9V	Diam
2.	0V	0V	5V	24,9V	Diam

No.	INT 1	INT 2	Vin	Vout	Kondisi Motor
3.	0V	5V	0V	24,9V	Diam
4.	5V	0V	0V	24,9V	Diam
5.	0V	5V	5V	24,9V	Berputar <i>Forward</i>
6.	5V	0V	5V	24,9V	Berputar <i>Reverse</i>
7.	5V	5V	0V	24,9V	Diam
8.	5V	5V	5V	24,9V	Diam

4.3 Pengujian Program Pembacaan *Rotary Encoder* pada DC Planet *Geared Motor*

Rotary Encoder yang dimaksudkan disini adalah *rotary encoder* pada DC Planet *Geared Motor*. Tujuan dari pengujian program pembacaan *rotary encoder* ini adalah untuk mengetahui apakah program yang dibuat sudah dapat diaplikasikan atau belum sehingga nantinya hasil *rotary encoder* dari program yang masuk akan sesuai dengan data *rotary encoder* pada motor yang sebenarnya.

Berikut ini adalah alur program yang digambarkan pada *flowchart*. Program ini dipergunakan untuk membaca putaran motor *rotary encoder*. Dimana untuk programnya sendiri dapat dilihat pada lampiran B.



Gambar 4.4. Flowchart Program Pembacaan Rotary Encoder

Pada DC Planet *Geared Motor Type* PG 36, diketahui bahwa :

PPR (*Pulse per Rotation*) DC Planet *Geared Motor* = 7
Gear Rotation = 26,9

Dan pada tugas akhir ini, *setting* dari program yang dipergunakan untuk pengambilan data serta pengujian menggunakan *any change interrupt mode* yang ter *setting* pada

software Code Vision AVR. Sehingga, nilai yang di *input* kan adalah 4 (*any change interrupt mode*).

Oleh karena itu, dapat dirumuskan nilai *Rotary Encoder* menurut *datasheet* dari motor :

$$\begin{aligned} & \text{PPR} \times \text{Gear Ratio} \times \text{Pengambilan Data} \\ &= 7 \times 26,9 \times 4 \\ &= 753,2 \\ &\approx 753 \end{aligned}$$

Nilai 753 yang dihasilkan dari perhitungan tersebut merupakan nilai *rotary encoder* motor pada putaran pertama. Maksudnya adalah ketika pemegang *knob* pada robot diputar sebanyak satu kali dari titik awal dan kembali pada titik tersebut lagi. Dimana pada putaran kedua, nilai $753 \times 2 = 1506$. Pada putaran ketiga, nilai $753 \times 3 = 2259$. Hal tersebut berulang - ulang untuk putaran keempat dan seterusnya.

Dari dua pengujian yang dibandingkan (pengujian program dan pengujian perhitungan) dapatkan dirumus *ERROR* sebagai berikut :

$$\text{Error} = \text{Rotary Encoder Real} - \text{Rotary Encoder Perhitungan}$$

Keterangan :

Rotary Encoder Real : Nilai dari *Rotary Encoder* pada motor yang sebenarnya saat memutar pemegang *knob*.

Rotary Encoder Perhitungan : Nilai dari *Rotary Encoder* pada motor menggunakan perhitungan dari *datasheet*

Berikut ini adalah hasil dari pengujian tersebut :

Tabel 4.2 Hasil Pengujian Program *Rotary Encoder*

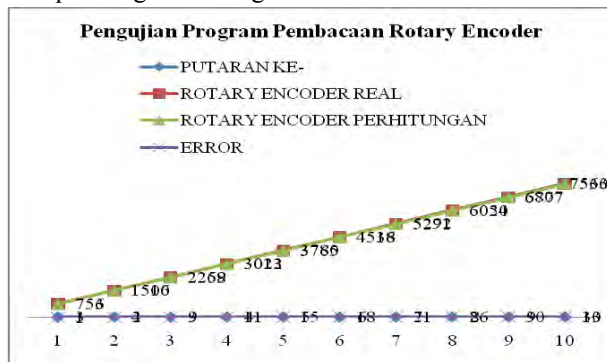
NO.	PUTARAN KE-	<i>ROTARY ENCODER REAL</i>	<i>ROTARY ENCODER PERHITUNGAN</i>	<i>ERROR</i>
1.	1	756	753	3
2.	2	1510	1506	4
3.	3	2268	2259	9

NO.	PUTARAN KE-	<i>ROTARY ENCODER REAL</i>	<i>ROTARY ENCODER PERHITUNGAN</i>	<i>ERROR</i>
4.	4	3023	3012	11
5.	5	3780	3765	15
6.	6	4536	4518	18
7.	7	5292	5271	21
8.	8	6050	6024	26
9.	9	6807	6777	30
10.	10	7563	7530	33

Keterangan :

- Putaran Ke- : Putaran pemegang *knob* yang dilakukan untuk pengujian (robot belum dipasang pada brankas)
- Rotary Encoder Real* : Nilai *rotary encoder* yang ditampilkan pada LCD selama melakukan putaran
- Rotary Encoder Perhitungan* : Data Perhitungan dari rumus ($PPR \times Gear\ Ratio \times \text{Pengambilan Data}$)
- Error* : Selisih dari Data *Rotary* dan Data *Perhitungan*

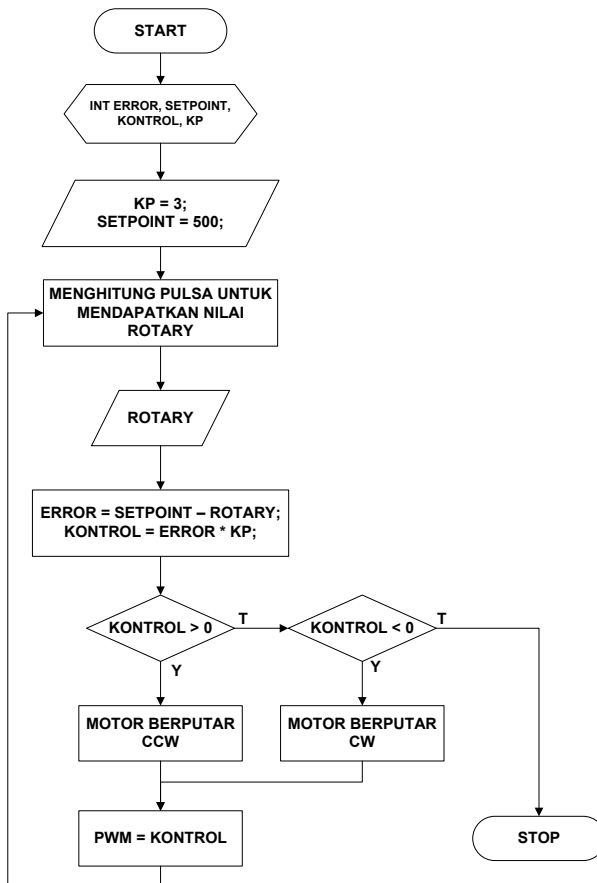
Dan didapatkan grafik sebagai berikut :



Gambar 4.5 Grafik Pengujian Program Pembacaan *Rotary Encoder*

4.4 Pengujian Program *Driver Motor* terhadap DC Planet *Geared Motor* dan Dibaca oleh Kontroler

Tujuan dari pengujian ini adalah untuk mengetahui apakah program yang digunakan sudah berjalan dan berfungsi sebagaimana yang diharapkan, Berikut ini adalah alur program yang digambarkan dalam *flowchart* yang dipergunakan untuk pengujian *driver motor*. Dimana program ini dapat dilihat pada Lampiran B.



Gambar 4.6. *Flowchart Program Driver Motor*

dari pengujian data di atas, didapatkan hasil sebagai berikut :

Tabel 4.3. Hasil Pengujian Program *Driver* Motor

Nilai yang Diinputkan pada Program Sebagai Nilai Awal	Nilai <i>Rotary</i> Motor yang Terbaca pada LCD	Nilai <i>Rotary</i> yang Diinginkan. Yang Dipergunakan Sebagai Acuan	<i>Error</i>
100	754	753	-1
220	1657	1657	0
300	2260	2259	+1
440	3314	3314	0
500	3766	3766	0

Keterangan :

Nilai yang Diinputkan pada Program Sebagai Nilai Awal : Nilai acuan atau nilai awal yang dipergunakan pada program *driver* motor

Nilai *Rotary* Motor yang terbaca pada LCD : Nilai *rotary* yang terbaca pada LCD saat dilakukan pengujian

Nilai *Rotary* yang Diinginkan. Yang Dipergunakan Sebagai Acuan : Data yang didapatkan dari (data yang diinputkan pada program x 7,53). 7,53 adalah *variabel* linierisasi yang didapat dari pengambilan data.

Error : Selisih dari Data *Rotary* yang terbaca pada LCD dengan data yang diinginkan sebagai acuan.

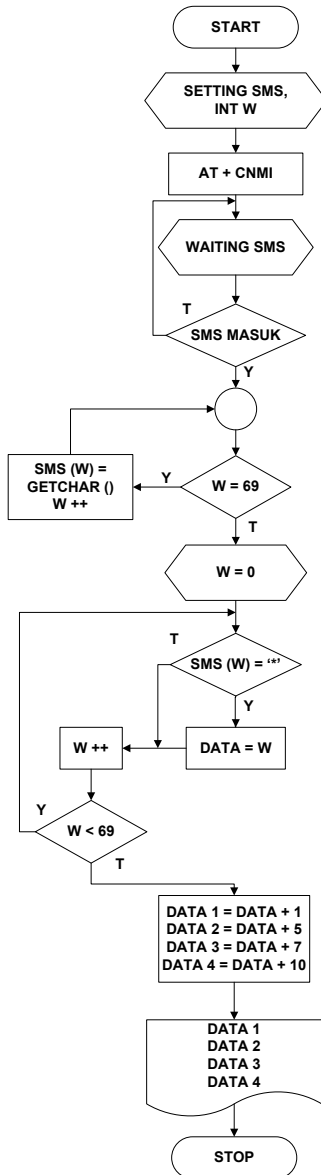
4.5 Pengujian Program Modul GSM SIM 900A terhadap Kontroler

Pengujian program modul GSM SIM 900A terhadap kontroler dilakukan dengan mengkonfigurasikannya menggunakan mikrokontroler ATmega 128. *Wiring* yang digunakan untuk menguji program modul GSM SIM 900A terhadap kontroler dijelaskan pada BAB 3.

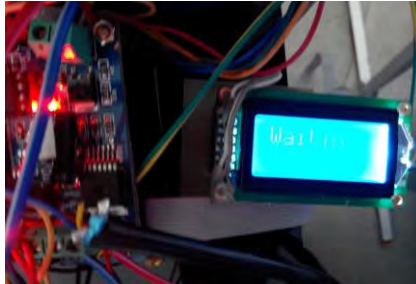
Pengujian dilakukan dengan cara membaca data yang masuk ke SIM 900A dilanjutkan ke kontroler. Data yang masuk ke SIM 900A adalah SMS (*Short Message Service*) yang dikirim oleh pemilik brankas melalui *handphonenya*. Data yang masuk akan langsung dibaca dan ditampilkan ke LCD. Disinilah akan diuji coba apakah saat penerimaan dan pembacaan *password*, terdapat data dari *password* yang hilang atau tidak. Data yang diambil adalah data yang telah *diparsing* sebelumnya di dalam program yang digunakan. Sehingga, pada LCD tidak akan menampilkan data mentah tanpa diolah terlebih dahulu.

Pada robot pembuka kunci brankas ini, format dari *password* brankas sudah ditentukan. Yaitu *00-00-XX-YY#OK. Dimana XX dan YY pada pada format tersebut merupakan kombinasi angka *password* brankas. Sebagai informasi, bahwa brankas yang dipergunakan pada tugas akhir ini hanya menggunakan dua kombinasi angka sebagai *password*-nya.

Berikut ini adalah alur dari program *parsing* yang dipergunakan. Untuk programnya dapat dilampirkan pada Lampiran B.

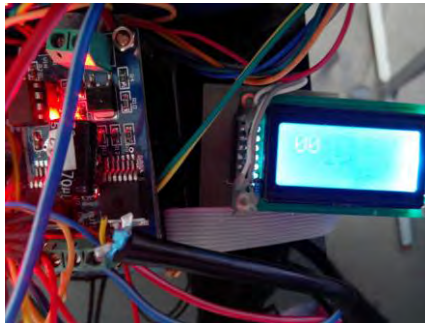


Gambar 4.7. Flowchart Program Modul GSM SIM 900A



Gambar 4.8. Pengujian Program Modul GSM SIM 900A (1)

Pada Gambar 4.7. Pengujian Program Modul GSM SIM 900A diatas diperlihatkan tampilan di LCD tertera "*waiting*" yang berarti Modul GSM SIM 900A sedang menunggu SMS yang akan dikirimkan oleh *user*.



Gambar 4.9 Pengujian Program Modul GSM SIM 900A (2)

Pada Gambar 4.8. Pengujian Program Modul GSM SIM 900A diatas terlihat tampilan pada LCD bertuliskan "00 00 10 30" yang berarti Modul tersebut sudah menerima SMS yang dikirim.

Yang mana SMS yang ditampilkan pada LCD tersebut sudah melalui *parsing* terlebih dahulu.

Berdasarkan pengambilan data yang telah diambil didapatkan hasil sebagai berikut :

Tabel 4.4. Hasil Pengujian Program Modul GSM SIM 900A

SMS <i>Password</i> yang dikirim melalui <i>Hand Phone</i>	Data SMS yang diterima oleh Modul GSM SIM 900A serta telah <i>diparsing</i> dan ditampilkan ke LCD			
	DATA 1	DATA 2	DATA 3	DATA 4
*00-00-10-30#OK	00	00	10	30
*12-00-12-33#OK	12	00	12	33
*22-12-10-30#OK	22	12	10	30
*12-00-10-30#OK	12	00	10	30
*23-00-00-30#OK	23	00	00	30
*11-22-33-44#OK	11	22	33	44
*123-00-10-303#OK	12	-0	-1	-3

4.6 Pengujian Kepresisian Putaran Motor

Tujuan dari pengujian kepresisian motor adalah untuk mengetahui apakah motor dapat berputar dan berhenti sesuai dengan data yang dimasukkan. Dengan acuan kepresisian putaran ini didapatkan dari *index* angka yang terdapat pada *knob* dimana angka tersebut diawali '0' sampai dengan '99'. Pada program untuk memutar motor, dianalogikan 10 kali putaran *knob* dan nilai yang diinputkan pada program adalah 1000. Apabila diinginkan motor berputar 5 kali maka data yang diinputkan adalah 500. Dan seterusnya untuk nilai - nilai berikutnya.

Berikut adalah hasil yang didapatkan dari beberapa pengujian yang telah dilakukan :

Tabel 4.5. Hasil Pengujian Kepresisian Putaran Motor

Nilai yang Diinputkan pada Program	<i>Index Knob</i> Awal (Saat Motor Belum berjalan)	Nilai <i>Rotary</i> Motor yang Dibaca pada LCD	<i>Index Knob</i> Setelah Motor Berjalan dan Kemudian Berhenti	Kondisi Putaran Motor
100	0	754	0	Motor CCW berputar satu kali putaran
-300	0	-2260	0	Motor CW berputar 3 kali putaran
200	0	1507	0	Motor CCW berputar 2 kali putaran
400	0	3013	0	Motor CCW berputar 4 kali putaran
600	0	4520	0	Motor CCW berputar 6 kali putaran
-50	0	-376	50	Motor CW berputar 1/2 putaran
40	0	302	40	Motor CW berputar 2/5 putaran

Keterangan :

- Nilai yang Diinputkan pada Program : Nilai tersebut berupa analogi. Dimana 100 berarti 1 kali putaran, 300 berarti 3 putaran, dan seterusnya
- Index Knob* Awal (Saat Motor Belum berjalan) : Robot dipasang pada pintu brankas. Yang mana *index knob* awal adalah saat *knob* brankas diposisikan pada nilai awal, yaitu 0
- Nilai *Rotary* Motor yang Dibaca pada LCD : Nilai tersebut merupakan nilai perhitungan yang perhitungannya sudah dijelaskan pada dua pengujian sebelumnya.

- Index Knob* Setelah Motor Berjalan dan Kemudian Berhenti : Pada saat posisi awal, *index knob* diposisikan pada suatu angka (misalnya 0). Kemudian motor menyala dan berputar menurut nilai yang diinputkan pada program. Dan ketika motor berhenti, maka jarum pada *knob* akan menunjuk pada sebuah nilai tertentu. Itulah yang dimaksudkan dengan *index knob* setelah motor berjalan dan kemudian berhenti.
- Kondisi Putaran Motor : Dmaksudkan untuk memperjelas putaran motor. Dimana ketika diinputkan suatu nilai pada program bagaimana putaran motor tersebut. Apakah CW atau CCW. Dan juga perhitungan banyaknya putaran yang dilakukan oleh motor.



Gambar 4.10. Pengujian Kepresisian Motor

4.7 Pengujian Alat Secara Keseluruhan

Pengujian alat secara keseluruhan menunjukkan bahwa robot pembuka brankas sudah berjalan tanpa gangguan dan berjalan sesuai fungsinya. Pengujian dilakukan secara bertahap. Cara pengujian alat secara keseluruhan untuk sekali proses pembukaan brankas adalah sebagai berikut :

1. Memasang robot pembuka brankas dengan tepat. Disesuaikan dengan *knob* brankas dan juga lengan brankas diatur serta holder penghisap dipasang dengan cara memutar penghisap tersebut. Hal ini harus dilakukan dengan benar dan tepat agar robot dapat merekat dengan kuat pada pintu brankas.



Gambar 4.11. Memasang Robot pada Pintu Brankas

2. Langkah selanjutnya adalah memposisikan *knob* pada posisi 0 (sesuai dengan kode putar brankas). Hal ini dikarenakan agar robot memiliki nilai awalan 0 sehingga nantinya motor dapat berputar presisi, sesuai dengan yang diharapkan.



Gambar 4.12. Memposisikan Knob pada Posisi 0

3. Berikutnya adalah dengan menutup bagian *security* pada brankas dengan menggunakan penutup yang terbuat dari *acrylic*. Pada penutup ini terdapat sensor *limit switch* yang berfungsi sebagai pengamanan sistem *security* pada brankas. Sebagai robot kemanan atau *safe security robot*, operator yang mengoperasikan robot tidak diperkenankan untuk melihat robot tersebut sedang memutar *knob* brankas. Oleh karena itu, robot pembuka brankas ini didesain harus terpasang penutup *knob* terlebih dahulu, baru robot ini akan bekerja. Apabila *knob* terbuka atau sengaja dibuka, maka sistem akan *error* dan robot akan mengirimkan SMS konfirmasi kepada *user* (pemilik brankas) bahwa sistem tidak dapat dijalankan.



Gambar 4.13. Menutup Bagian Security pada Brankas

4. Langkah selanjutnya adalah mengaktifkan *push button power* yang mana ketika *push button* aktif maka baterai akan aktif dan menyebabkan sistem menjadi aktif pula. Pada saat *push button* diaktifkan maka indikator LED juga akan aktif secara berurutan yang menandakan bahwa sistem sudah dapat dijalankan.



Gambar 4.14. Mengaktifkan Push Button Power

5. Setelah *push button power* diaktifkan dan LED indikator berhenti berkedip, langkah berikutnya adalah menekan tombol start pada *push button* berwarna merah di bawah *push button* kuning. Ketika *push button* merah diaktifkan maka tampilan LCD adalah "*waiting*" yang berarti Modul GSM SIM 900A sedang menunggu SMS dari *user* (pemilik brankas). Pengiriman SMS ke nomor modul SIM 900A (085755298956) dengan format *00-00-XX-XX*. Dimana XX ini adalah kode dari brankas tersebut. Selanjutnya setelah kode SMS diterima maka tampilan di LCD menjadi "*running system*" yang mana berarti sistem sudah dapat dioperasikan.



Gambar 4.15. Mengaktifkan Push Button Start

6. Selanjutnya motor akan berputar sesuai dengan kode putar yang diterima oleh Modul GSM SIM 900A.
7. Dan ketika motor sudah berhenti, dan brankas dapat dibuka maka Modul GSM SIM 900A akan mengirimkan SMS konfirmasi kepada nomor *user* (pemilik brankas) bahwa sistem sudah diselesaikan dengan baik dan brankas sudah terbuka.



Gambar 4.16. Brankas Sudah Terbuka



Gambar 4.17. Brankas Terbuka dan *User* Menerima SMS Konfirmasi

Tabel 4.6. menunjukkan data hasil pengujian robot pembuka brankas secara keseluruhan. Dari pengujian alat keseluruhan didapatkan hasil bahwa semua perangkat yang digunakan dapat bekerja normal sesuai dengan fungsinya.

Tabel 4.6. Hasil Pengujian Alat Secara Keseluruhan

Parameter	Status
<i>Push Button Power (Start Sistem)</i>	Mengaktifkan baterai dan sistem pada robot pembuka brankas
<i>LED 5 buah</i>	Indikator sistem sudah <i>ready</i> untuk digunakan
<i>Push Button I (Reset)</i>	Menghentikan sistem sehingga sistem akan <i>refresh</i> .
<i>Push Button II (Start SMS)</i>	Sistem sudah <i>ready</i> untuk menerima SMS dari <i>user</i>
<i>Limit Switch</i>	Mengamankan <i>security</i> kode
<i>Reed Switch 1 (S1)</i>	Mendeteksi medan magnet
<i>Reed Switch 2 (S2)</i>	Mendeteksi medan magnet
Modul GSM SIM 900A	Menerima SMS kode putar dari <i>user</i> dan mengirim SMS konfirmasi kepada <i>user</i>

Pada pengujian secara keseluruhan dijelaskan bahwa untuk sekali proses pembukaan brankas membutuhkan waktu selama 3 menit 33 detik. Foto alat keseluruhan dapat dilihat pada Lampiran A (halaman A-1), serta foto proses pengujian secara keseluruhan ditunjukkan pada Gambar 4.18. Data hasil waktu pengujian pembukaan brankas ditunjukan pada Tabel 4.7.

Tabel 4.7 Hasil Waktu Pengujian Alat secara Keseluruhan

Pengujian	Waktu
1	4 menit 14 detik
2	3 menit 53 detik
3	3 menit 24 detik



Gambar 4.18. Pengujian Secara Keseluruhan

BAB V

PENUTUP

Bab penutup ini berisi tentang kesimpulan yang diperoleh selama proses pembuatan robot pembuka brankas, kesimpulan dari hasil pengujian dan analisa data, serta saran untuk robot pembuka brankas ini kedepannya.

5.1 Kesimpulan

1. Nilai kecepatan putar maksimal yang diinputkan pada program adalah < 22 . Dimana data tersebut didapatkan dari pengujian yang telah dilakukan. Jika nilai kecepatan putar maksimal bernilai > 22 (batas maksimal dari kecepatan putar) maka motor akan berhenti tidak pada nilai yang menjadi *setpoint* (nilai awal putaran).
2. Tegangan kerja yang dipergunakan untuk *supply* pada modul GSM SIM 900A adalah sebesar 8-9 Volt. Dimana *supply* yang dipergunakan adalah baterai lipo 6 *cell* dengan kapasitas sebesar 2650 mAH. Jika dipergunakan secara terus menerus untuk membuka brankas, baterai akan tahan sampai 12 kali kerja.
3. Waktu yang diperlukan mulai dari pemasangan robot sampai dengan brankas terbuka (keseluruhan) adalah 3 menit 33 detik. Dimana waktu yang diperlukan robot untuk mengirimkan SMS kepada *user* sampai *user* menerima SMS adalah selama 11 detik. Sedangkan waktu eksekusi proses pembukaan brankas itu sendiri adalah 2 menit 25 detik.

5.2 Saran

1. Penggunaan LCD pada robot pembuka brankas ini seharusnya ditanggalkan. Agar lebih *secure*. Alangkah baiknya penggunaan LCD digantikan dengan penggunaan LED sebagai indikator

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] *Locks, Safes, and Security: An International Police Reference*, published by Charles Thomas Publishers, Springfield, Illinois, United States. (2000)
- [2] Datasheet *Microcontroller* ATmega 128
- [3] Rizqiawan Arwindra, "**Sekilas Rotary Encoder**", <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>, 10 Mei 2015
- [4] Eitel, Elisabeth. "**Basics of Rotary Encoders: Overview and New Technologies**" / *Machine Design Magazine*, 7 May 2014
- [5] SIM900 AT *Command Manual*
- [6] Harald Naunheimer; Peter Fietkau; G Lechner. "**Automotive transmissions : fundamentals, selection, design and application (2nd ed.)**". 2011
- [7] Iksan, "**Pengertian Gearbox**", <http://fungsi.info/fungsi-gearbox-pada-kendaraan/>, 17 Mei 2015
- [8], "**Power Motor Driver Module**", http://www.alibaba.com/product-detail/BTS7971B-BTN7971B-power-motor-driver-module_1630601713.html. 28 April 2015
- [9] Spitzer, David W. "**Variable Speed Drives. Instrument Society of America**". 1990
- [10] K. Kondo, H. Terao, H. Abe, M. Ohta, K. Suzuki, T. Sasaki, G. Kawachi, J. Ohwada, "**Liquid Crystal Display Device**", filed Sep 18, 1992 and Jan 20, 1993.
- [11] Rudolf F. Graf, "**Reed Relay**" Dictionary of Electronics; Radio Shack, 1974-75. Fort Worth, Texas.
- [12], "**Reed Switch**", <http://indo-ware.com/produk-238-reed-switch.html>. 2 Mei 2015
- [13] Stephen Herman, "**Industrial Motor Control Cengage Learning, chapter 11 Limit Switches**". 2011

-----Halaman ini sengaja dikosongkan-----

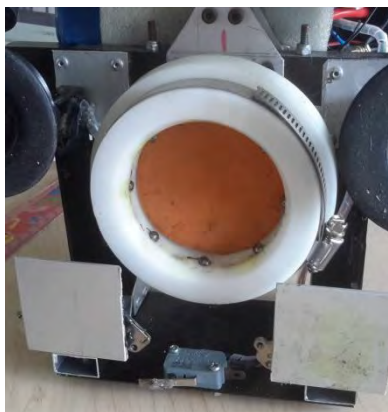
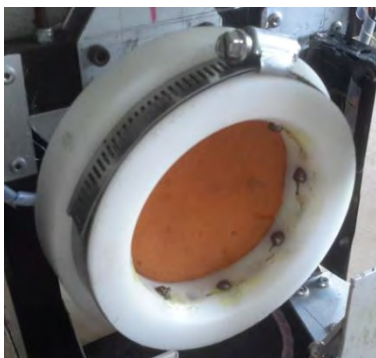
LAMPIRAN A FOTO

Robot Pembuka Brankas



Perancangan Mekanik Robot

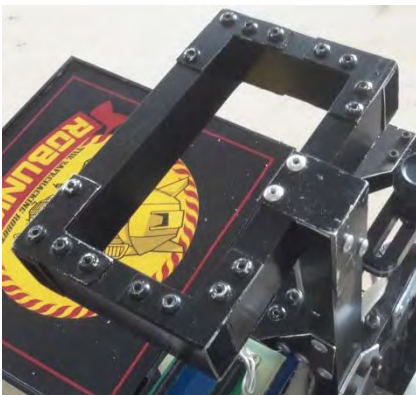
1. Pemegang *Knob*

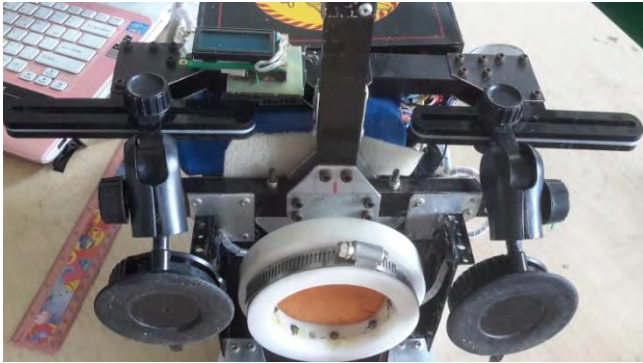


2. Casing Elektronik

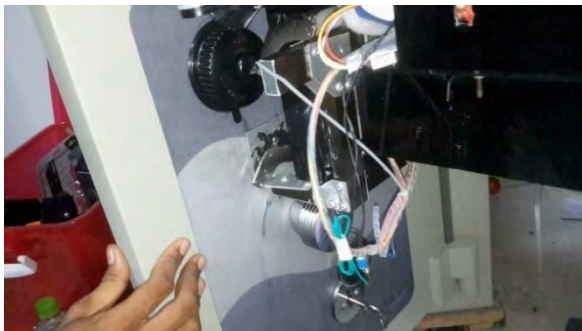
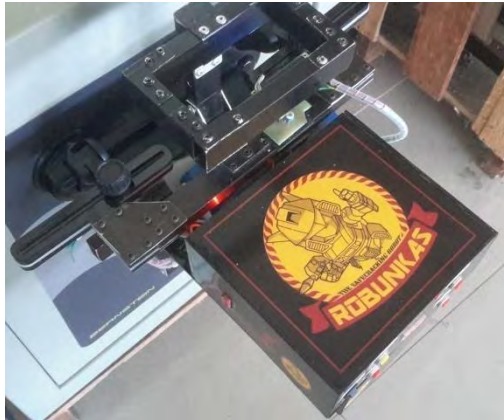


3. Bagian Inti Robot

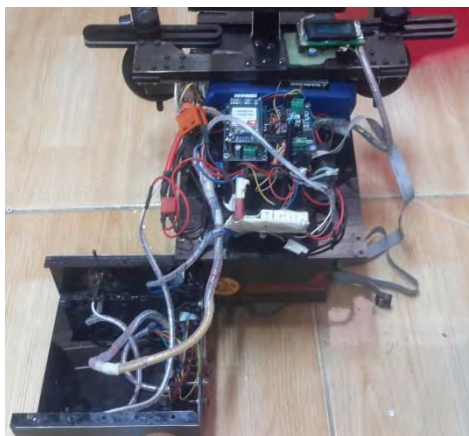
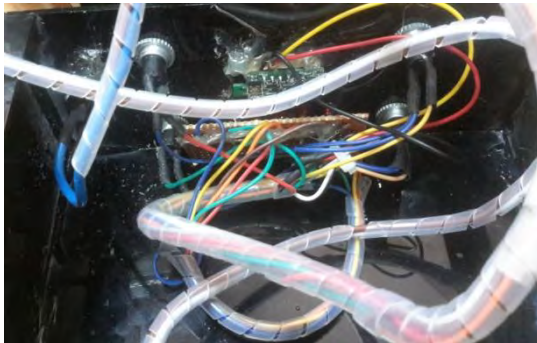
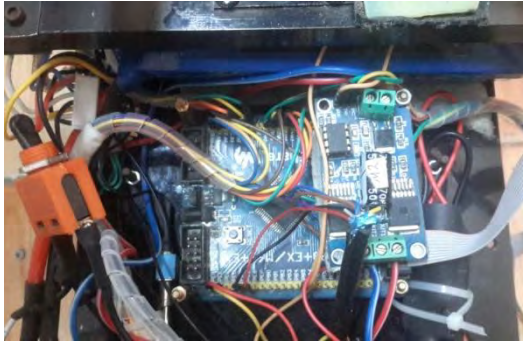




4. Pengujian Secara Keseluruhan



Bagian Elektronik pada Robot



LAMPIRAN B PROGRAM

/*****

Project : Tugas Akhir
Version : 0.1
Date : 3/8/2015
Author : Alin Wahyu Setiawan (2212039043)
Author 2 : Shofia Kholisatus Sa'adah (2212039002)
Counsellor : Rudy Dikairono, S.T., M.T
Laboratory : Center Of Robotics Sepuluh Nopember
Company : Institut Teknologi Sepuluh Nopember

Chip type : ATmega128
Program type : Application
AVR Core Clock frequency: 16.000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 1024

*****/

```
#include <mega128.h>
#include <delay.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <alcd.h>
```

```
//sensor penutup
//kuning pinF.5
//
//
//
unsigned char sinyal[30];
unsigned char pulsa[100];
```

```

unsigned                                                    char
no1[100],no2[100],no3[100],no4[100],no5[100],no6[100],no7[100],no8
[100],no9[100],no10[100],no11[100],no12[255],no13[255];
unsigned                                                    int
datano1,datano2,datano3,datano4,datano5,datano6,datano7,datano8,data
no9,datano10,datano11,datano12,datano13;

```

```

unsigned int n,w,x,a,c,z,data,data1,data2,data3,data4,datano;
unsigned int ram_bebas,indexbuf;
unsigned                                                    char
buf[360],take_char[360],buffer[10],buf1[10],buf2[10],buf3[10],buf4[10]
,buf5[255];
unsigned char bufnomor[255];

```

```

int i;
int status = 0;
int LastRotary;
int FirstError;
int maks = 19; //11
int mini = 15; //8
float nilaibagi = 7.532;
int index=1,index2,step=0;
int nilai1 = 350;
int nilai2 = 200;
int rotary[2] = {0, 0};
char temp[16],temp2[16],temp3[16];
char nohp[14]={'+','6','2','8','5','6','4','6','4','5','0','6','5','0'};

```

```

// Declare your global variables here
#define kiri 0
#define kiri2 1
#define UART_NO_DATA      0x0100      /* no receive data
available */
#define PWM OCR1AL

```

```

//declare pin lampu
#define lampu1 PORTA.2
#define lampu2 PORTA.3

```

```

#define lampu3 PORTA.4
#define lampu4 PORTA.5
#define lampu5 PORTA.6

// External Interrupt 4 service routine
interrupt [EXT_INT4] void ext_int4_isr(void)
{
// Place your code here
if(PINE.4==PINE.5)
    rotary[kiri]++;
    else
    rotary[kiri]--;
}

// External Interrupt 5 service routine
interrupt [EXT_INT5] void ext_int5_isr(void)
{
// Place your code here
if(PINE.4==PINE.5)
    rotary[kiri]--;
    else
    rotary[kiri]++;

}

void m_cw(){
PORTA.0 = 1;
PORTA.1 = 0;
}

void m_ccw(){
PORTA.0 = 0;
PORTA.1 = 1;
}

void m_stop(){
PORTA.0 = 0;
PORTA.1 = 0;
PWM =0;
}

```

```

}

void hitung_pulsa(){
rotary[kiri]=((rotary[kiri]));
}

void cek_sinyalhp(){
//provider im3
printf("AT+CSQ");putchar(13);
lcd_clear();lcd_gotoxy(0,0);lcd_putsf("Waiting");
w=0;
while(w != 20){
take_char[w] = getchar();
w++;
}
i=10;
lcd_clear();lcd_gotoxy
(0,0);sprintf(sinyal,"%c%c%c%c%c%c%c",take_char[i],take_char[i+1],take
_char[i+2],take_char[i+3],take_char[i+4],take_char[i+5]);lcd_puts(siny
al);
//delay_ms(300);

}

void cek_pulsaahp(){
//provider im3
printf("ATD*388#");putchar(13);
lcd_clear();lcd_gotoxy(0,0);lcd_putsf("Waiting");
w=0;
while(w != 60){
take_char[w] = getchar();
w++;
}
i=34 ;
lcd_clear();lcd_gotoxy(0,0);sprintf(pulsa,"Sisa Pulsa Anda :
%c%c%c%c%c%c%c dan Masa Aktif Anda :
%c%c%c%c%c%c%c%c",take_char[i],take_char[i+1],take_char[i+2],ta
ke_char[i+3],take_char[i+4],take_char[i+5],take_char[i+12],take_char[i

```

```

+13],take_char[i+14],take_char[i+15],take_char[i+16],take_char[i+17],t
ake_char[i+18],take_char[i+19]);lcd_puts(pulsa);
delay_ms(300);
    for(w=0; w<=60; w++){
        take_char[w] = 0;
    }

```

```

void cek_modem(){
cek_pulsa();
cek_sinyal();
}

```

```

void bacalagi(){
lcd_clear();lcd_gotoxy (0,0);lcd_putsf("Waiting");
delay_ms(100);
w=0;
printf ("AT+CNMI=2,2");putchar (13);while (getchar() != 'K'){
while (w != 69){
//while (getchar() != 'K'){
    take_char[w] = getchar();
    w++;
}
    lcd_clear();delay_ms(1000);

```

```

for(w=0; w<=69; w++){
    if(take_char[w]=='*'){
        data=w;
    }
}
}

```

```

n=11 ;
sprintf(no1,"%c",take_char[n+1]);
sprintf(no2,"%c",take_char[n+2]);
sprintf(no3,"%c",take_char[n+3]);
sprintf(no4,"%c",take_char[n+4]);
sprintf(no5,"%c",take_char[n+5]);
sprintf(no6,"%c",take_char[n+6]);

```

```

sprintf(no7,"%c",take_char[n+7]);
sprintf(no8,"%c",take_char[n+8]);
sprintf(no9,"%c",take_char[n+9]);
sprintf(no10,"%c",take_char[n+10]);
sprintf(no11,"%c",take_char[n+11]);
sprintf(no12,"%c",take_char[n+12]);
sprintf(no13,"%c",take_char[n+13]);
datano1 = atoi(no1);
datano2 = atoi(no2);
datano3 = atoi(no3);
datano4 = atoi(no4);
datano5 = atoi(no5);
datano6 = atoi(no6);
datano7 = atoi(no7);
datano8 = atoi(no8);
datano9 = atoi(no9);
datano10 = atoi(no10);
datano11 = atoi(no11);
datano12 = atoi(no12);
datano13 = atoi(no13);

```

```

sprintf(buf1,"%c%c",take_char[data+1],take_char[data+2]);
sprintf(buf2,"%c%c",take_char[data+4],take_char[data+5]);
sprintf(buf3,"%c%c",take_char[data+7],take_char[data+8]);
sprintf(buf4,"%c%c",take_char[data+10],take_char[data+11]);

```

```

data1=atoi(buf1);
data2=atoi(buf2);
data3=atoi(buf3);
data4=atoi(buf4);
lcd_clear();lcd_gotoxy(0,0);sprintf(buf,"%d,%d,%d          dan
%d",data1,data2,data3,data4);lcd_puts(buf);delay_ms(5000);
//lcd_clear();lcd_gotoxy(0,0);sprintf(bufnomor,"+%d%d%d%d%d%d%d%
d%d%d%d%d%d",datano1,datano2,datano3,datano4,datano5,datano
6,datano7,datano8,datano9,datano10,datano11,datano12,datano13);lcd_
puts(bufnomor);delay_ms(5000);
//mengirim sms lagi

```

```

step=step+1;

```

```

}

void bacalagibuatdata(){
lcd_clear();lcd_gotoxy(0,0);lcd_putsf("Waiting");
delay_ms(1000);
w=0;
printf ("AT+CNMI=2,2");putchar (13);while (getchar() != 'K'){
while (w != 69){
//while (getchar() != 'K'){
take_char[w] = getchar();
w++;
}
lcd_clear();delay_ms(1000);

for(w=0; w<=69; w++){
if(take_char[w]=='*'){
data=w;
}
}

sprintf(buf1,"%c%c",take_char[data+1],take_char[data+2]);
sprintf(buf2,"%c%c",take_char[data+4],take_char[data+5]);
sprintf(buf3,"%c%c",take_char[data+7],take_char[data+8]);
sprintf(buf4,"%c%c",take_char[data+10],take_char[data+11]);
data1=atoi(buf1);
data2=atoi(buf2);
data3=atoi(buf3);
data4=atoi(buf4);
lcd_clear();lcd_gotoxy(0,0);lcd_puts(buf1);delay_ms(2000);
lcd_clear();lcd_gotoxy(0,0);lcd_puts(buf2);delay_ms(2000);
lcd_clear();lcd_gotoxy(0,0);lcd_puts(buf3);delay_ms(2000);
lcd_clear();lcd_gotoxy(0,0);lcd_puts(buf4);delay_ms(2000);
lcd_clear();

}

void tes_motor(){
PWM =10;

```



```

if(PINF.2==0 && PINF.0==0){
PWM=6;
m_cw();
}
else if(PINF.2==0 && PINF.1==0){
PWM=6;
m_ccw();
}
else if(PINF.3==0 && PINF.1==0){
PWM=15;
m_ccw();
}
else if(PINF.3==0 && PINF.0==0){
PWM=15;
m_cw();
}
else if(PINF.0==0){
m_cw();
}
else if(PINF.1==0){
m_ccw();
}
else {
m_stop();
}
//OCR1AL =0;
hitung_pulsa();
ftoa(rotary[kiri],1,temp);
ftoa(index2,1,temp2);
lcd_gotoxy(0,0);lcd_puts(temp);
lcd_gotoxy(0,1);lcd_puts(temp2);
}

void baca_sms(){
lcd_clear();lcd_gotoxy (0,0);lcd_putsf("Waiting Read SMS");
delay_ms(100);
w=0;

printf ("AT+CNMI=2,2");putchar (13);while (getchar() != 'K'){}
```

```

// printf("AT+CMGR=1"); putchar(13);
while (w != 69){
//while (getchar() != 'K'){
    take_char[w] = getchar();
    w++;
}
lcd_clear();
x=0;
for(w=0; w<16; w++){
    lcd_gotoxy(w,0);
    sprintf(buf,"%c",take_char[x]);
    lcd_puts(buf);
    x++;
}
x=16;
for(w=0; w<16; w++){
    lcd_gotoxy(w,1);
    sprintf(buf,"%c",take_char[x]);
    lcd_puts(buf);
    x++;
}
delay_ms(5000);
lcd_clear();
x=32;
for(w=0; w<16; w++){
    lcd_gotoxy(w,0); sprintf(buf,"%c",take_char[x]);
    lcd_puts(buf);
    x++;
}
x=48;
for(w=0; w<16; w++){
    lcd_gotoxy(w,1);
    sprintf(buf,"%c",take_char[x]);
    lcd_puts(buf);
    x++;
}
delay_ms(5000);
lcd_clear();
x=64;

```

```

for(w=0; w<16; w++){
    lcd_gotoxy (w,0);
    sprintf(buf,"%c",take_char[x]);
    lcd_puts(buf);
    x++;
}
x=80;
for(w=0; w<16; w++){
    lcd_gotoxy (w,1);
    sprintf(buf,"%c",take_char[x]);
    lcd_puts(buf);
    x++;
}
delay_ms(5000);
lcd_gotoxy (0,0);lcd_putsf("Waiting Read SMS");
}

void kirim_sms(int command){
    lcd_clear();lcd_gotoxy(0,0);lcd_putsf("Sending SMS");
    switch (command) {
        case 1:
            printf ("AT+CMGS=");
            putchar                                     ("");printf
("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c",nohp[0],nohp[1],nohp[2
],nohp[3],nohp[4],nohp[5],nohp[6],nohp[7],nohp[8],nohp[9],nohp[10],n
ohp[11],nohp[12],nohp[13]);putchar ("");
            putchar (13);delay_ms (1000);
            printf ("Sistem telah diselesaikan dengan baik %s",pulsa);
            putchar (26);
            //putchar(10);
            putchar (13);
            //while (getchar() != 'K') {}
            delay_ms(100);
        break;
        case 2:
            printf ("AT+CMGS=");
            putchar                                     ("");printf
("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c",nohp[0],nohp[1],nohp[2

```

```

],nohp[3],nohp[4],nohp[5],nohp[6],nohp[7],nohp[8],nohp[9],nohp[10],nohp[11],nohp[12],nohp[13]);putchar ("");
    putchar (13);delay_ms (1000);
    printf ("Sistem Error!!! Kemungkinan Robot telah DI
MANIPULASI. SEMUA SISTEM ROBOT DIMATIKAN UNTUK
MENJAGA KEAMANAN DARI PASSWORD BRANKAS");
    putchar (26);
    //putchar(10);
    putchar (13);
    //while (getchar() != 'K'){
    delay_ms(100);

break;
case 3:
    printf ("AT+CMGS=");
    putchar ("");printf
("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c",nohp[0],nohp[1],nohp[2
],nohp[3],nohp[4],nohp[5],nohp[6],nohp[7],nohp[8],nohp[9],nohp[10],nohp[11],nohp[12],nohp[13]);putchar ("");
    putchar (13);delay_ms (1000);
    printf ("Perintah telah dinyalakan , Balas dengan mengirimkan
password anda.");
    putchar (26);
    //putchar(10);
    putchar (13);
    //while (getchar() != 'K'){
    delay_ms(100);

break;
default:
};

}

```

```

void jalan_motor(){
hitung_pulsa();
switch (index) {

```

```

case 1:
    while (rotary[kiri] <= nilai1){ //nilai1=350
        m_ccw();
        PWM = 5;
        ftoa(rotary[kiri],1,temp);lcd_gotoxy(0,0);lcd_puts(temp);
    }
    m_stop();
    delay_ms(2000);
    index = 2;
    break;
case 2:
    while (rotary[kiri] >= nilai2){ //nilai2=200
        m_cw();
        PWM = 5;
        ftoa(rotary[kiri],1,temp);lcd_gotoxy(0,0);lcd_puts(temp);
    }
    m_stop();
    delay_ms(2000);
    index = 3;
    break;
case 3:
    lcd_clear();
    break;

default:
    };
}

```

```

void kontrol_posisi(int angka){
int Error,Setpoint,kontrol,KP,maks=11;
KP = 3;
Setpoint=angka;
hitung_pulsa();
Error = Setpoint - rotary[kiri];
kontrol = (Error*KP);

```

```

if (FirstError <= 0){
    if (Error <= (FirstError-(FirstError/20))){
        maks = maks;
    }
}

```

```

}
else if (Error <= (FirstError-(FirstError/19))) {
    maks = maks+1;
}
else if (Error <= (FirstError-(FirstError/18))) {
    maks = maks+2;
}
else if (Error <= (FirstError-(FirstError/17))) {
    maks = maks+3;
}
else if (Error <= (FirstError-(FirstError/16))) {
    maks = maks+4;
}
else if (Error <= (FirstError-(FirstError/15))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/14))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/13))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/12))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/11))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/10))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/9))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/8))) {
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/7))) {
    maks = maks+5;
}

```

```

}
else if (Error <= (FirstError-(FirstError/6))){
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/5))){
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/4))){
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/3))){
    maks = maks+5;
}
else if (Error <= (FirstError-(FirstError/2))){
    maks = maks+5;
}
else if (Error <= (FirstError/2)){
    maks = maks+5;
}
else if (Error <= (FirstError/3)){
    maks = maks+5;
}
else if (Error <= (FirstError/4)){
    maks = maks+5;
}
else if (Error <= (FirstError/5)){
    maks = maks+5;
}
else if (Error <= (FirstError/6)){
    maks = maks+5;
}
else if (Error <= (FirstError/7)){
    maks = maks+5;
}
else if (Error <= (FirstError/8)){
    maks = maks+5;
}
else if (Error <= (FirstError/9)){
    maks = maks+5;
}

```

```

}
    else if (Error <= (FirstError/10)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/11)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/12)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/13)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/14)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/15)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/16)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/17)){
        maks = maks+5;
    }
    else if (Error <= (FirstError/18)){
        maks = maks+4;
    }
    else if (Error <= (FirstError/19)){
        maks = maks+3;
    }
    else if (Error <= (FirstError/20)){
        maks = maks+2;
    }
    else{
        maks = maks+1;
    }
}
}
else{

```



```

    if (Error >= (FirstError-(FirstError/20))) {
        maks = maks;
    }
    else if (Error >= (FirstError-(FirstError/19))) {
        maks = maks+1;
    }
    else if (Error >= (FirstError-(FirstError/18))) {
        maks = maks+2;
    }
    else if (Error >= (FirstError-(FirstError/17))) {
        maks = maks+3;
    }
    else if (Error >= (FirstError-(FirstError/16))) {
        maks = maks+4;
    }
    else if (Error >= (FirstError-(FirstError/15))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/14))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/13))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/12))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/11))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/10))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/9))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/8))) {
        maks = maks+5;
    }

```

```

    else if (Error >= (FirstError-(FirstError/7))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/6))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/5))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/4))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/3))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError-(FirstError/2))) {
        maks = maks+5;
    }
    else if (Error >= (FirstError/2)) {
        maks = maks+5;
    }
    else if (Error >= (FirstError/3)) {
        maks = maks+5;
    }
    else if (Error >= (FirstError/4)) {
        maks = maks+5;
    }
    else if (Error >= (FirstError/5)) {
        maks = maks+5;
    }
    else if (Error >= (FirstError/6)) {
        maks = maks+5;
    }
    else if (Error >= (FirstError/7)) {
        maks = maks+5;
    }
    else if (Error >= (FirstError/8)) {
        maks = maks+5;
    }

```

```

    else if (Error >= (FirstError/9)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/10)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/11)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/12)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/13)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/14)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/15)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/16)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/17)){
        maks = maks+5;
    }
    else if (Error >= (FirstError/18)){
        maks = maks+4;
    }
    else if (Error >= (FirstError/19)){
        maks = maks+3;
    }
    else if (Error >= (FirstError/20)){
        maks = maks+2;
    }
    else{
        maks = maks+1;
    }

```

```
}
```

```
if(kontrol>maks){ //untuk membatasi nilai maksimal kontrol atau PWM
```

```
    kontrol =maks;  
}
```

```
else if(kontrol<(-1*maks)){  
    kontrol =(-1*maks);  
}
```

```
if(kontrol<=mini && kontrol>0){ //untuk membatasi nilai minimum  
    kontrol =mini;  
}
```

```
if(kontrol>=((-1)*mini) && kontrol<0){  
    kontrol =((-1)*mini);  
}
```

```
if (kontrol>0){ //untuk menentukan arah  
m_ccw();  
}
```

```
if (kontrol<0){  
m_cw();  
kontrol=kontrol*-1;  
}
```

```
PWM= kontrol;
```

```
    ftoa(rotary[kiri],1,temp);lcd_gotoxy(0,0);lcd_puts(temp);  
    ftoa(kontrol,1,temp2);lcd_gotoxy(0,1);lcd_puts(temp2);  
    ftoa(FirstError,1,temp3);lcd_gotoxy(0,2);lcd_puts(temp3);
```

```
}
```

```
void cek_tombol(){  
if(PINF.0==0){  
    lcd_clear();  
    lcd_gotoxy(0,0);lcd_putsf("OK");  
    lcd_gotoxy(0,1);lcd_putsf("1");
```

```

    }
    else if(PINF.1==0){
        lcd_clear();
        lcd_gotoxy(0,0);lcd_putsf("OK");
        lcd_gotoxy(0,1);lcd_putsf("2");
    }
    else if(PINF.2==0){
        lcd_clear();
        lcd_gotoxy(0,0);lcd_putsf("OK");
        lcd_gotoxy(0,1);lcd_putsf("3");
    }
    else if(PINF.3==0){
        lcd_clear();
        lcd_gotoxy(0,0);lcd_putsf("OK");
        lcd_gotoxy(0,1);lcd_putsf("4");
    }
    else if(PINF.4==0){
        lcd_clear();
        lcd_gotoxy(0,0);lcd_putsf("OK");
        lcd_gotoxy(0,1);lcd_putsf("5");
    }
    else if(PINF.5==0){
        lcd_clear();
        lcd_gotoxy(0,0);lcd_putsf("OK");
        lcd_gotoxy(0,1);lcd_putsf("6");
    }

}

//void putar2(){
//int nilai,kontrol; //skala 1000
//nilai = (nilai*7.510); //nilai pengali satu putaran 7.517
//if(PINF.0==0){
//kontrol_posisi(nilai);
//    }
//    kontrol =PWM;

```

```

//lcd_clear();
//  ftoa(rotary[kiri],1,temp);lcd_gotoxy(0,0);lcd_puts(temp);
//  ftoa(kontrol,1,temp2);lcd_gotoxy(0,1);lcd_puts(temp2);
//  ftoa(PWM,1,temp3);lcd_gotoxy(0,2);lcd_puts(temp3);
//}

void cek_sensor(){
  //sensor
  if ((PINF.4==0)&&(PINF.3==0) && (PINF.5==0)) {
    status = 1;

  }
  else{
    status = 0;
  }
}

void buka_brankas( int nilai){
  int kontrol; //skala 1000
  hitung_pulsa();
  nilai =ceil((nilai*nilaibagi)); //7.573 //7.517 //7482
  kontrol_posisi(nilai);
  kontrol =PWM;
  lcd_clear();

  ftoa(ceil((nilai)),1,temp2);lcd_gotoxy(0,0);lcd_puts(temp2);
  //ftoa(PWM,1,temp3);lcd_gotoxy(0,1);lcd_puts(temp3);
  if(PWM<=0){
    LastRotary = rotary[kiri];
    step = step +1;
    delay_ms(5000);
  }
}

void final(){
  cek_modem();
}

```

```

    m_cw();
    PWM = 15;
    delay_ms(600);
    kirim_sms(1);
    m_stop();
    PWM = 15;
    delay_ms(25000);
    m_ccw();
    delay_ms(10000);
    m_stop();
    delay_ms(1000000);
}

```

```

void cek_koneksi(){
    lampu5 = 1;
    printf("ATE0");putchar(13);while (getchar() != 'K'){
    lampu1 = 1;
    printf("AT");putchar(13);while (getchar() != 'K'){
    lampu2 = 1;
    printf("AT+CMGF=1");putchar(13);while (getchar() != 'K'){
    delay_ms(500);
    lcd_clear();
    }
}

```

```

void main(void)
{

```

```

// Declare your local variables here

```

```

// Input/Output Ports initialization

```

```

// Port A initialization

```

```

// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=Out
Func1=Out Func0=Out

```

```

// State7=T State6=T State5=T State4=T State3=0 State2=0 State1=0
State0=0

```

```

PORTA=0x00;

```

```

DDRA=0xFF;

```

```

// Port B initialization

```

```

// Func7=In Func6=Out Func5=Out Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=0 State5=0 State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x60;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x0F;
DDRD=0x30;

// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTE=0x00;
DDRE=0x00;

// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTF=0xFF;
DDRF=0x00;

```



```

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 250.000 kHz
// Mode: Ph. correct PWM top=0x00FF
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0xA1;
TCCR1B=0x02;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;

```

```
OCR1BL=0x00;  
OCR1CH=0x00;  
OCR1CL=0x00;
```

```
// External Interrupt(s) initialization  
// INT0: Off  
// INT1: Off  
// INT2: Off  
// INT3: Off  
// INT4: On  
// INT4 Mode: Any change  
// INT5: On  
// INT5 Mode: Any change  
// INT6: Off  
// INT7: Off
```

```
//EICRA=0x00; //int4 saja  
//EICRB=0x01;  
//EIMSK=0x10;  
//EIFR=0x10;
```

```
EICRA=0x00; //int 4 dan int5  
EICRB=0x05;  
EIMSK=0x30;  
EIFR=0x30;
```

```
//EICRA=0x00; //int non aktif  
//EICRB=0x00;  
//EIMSK=0x00;
```

```
// USART0 initialization  
// Communication Parameters: 8 Data, 1 Stop, No Parity  
// USART0 Receiver: On  
// USART0 Transmitter: On  
// USART0 Mode: Asynchronous  
// USART0 Baud Rate: 9600
```

```

UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x67;

// USART1 initialization
// USART1 disabled
UCSR1B=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 12

```

```

// Global enable interrupts
#asm("sei")
lcd_init(8); //LCD kolom
lcd_clear();
//setting baud rate
lampu3 = 1;
delay_ms(100);
lcd_gotoxy(0,0);lcd_putsf("Loading");
lcd_gotoxy(0,0);lcd_putsf("Setting");
delay_ms(5000);
cek_koneksi();
lcd_gotoxy(0,0);lcd_puts("Starting");
delay_ms(1000);
lampu4 = 1;
delay_ms(500);

lampu1=0;lampu2=0;lampu3=0;lampu4=0;lampu5=0;
while (1)
{

    lcd_clear();
    cek_sensor();

    if(status == 1){

    if (PINF.0==0){
        kirim_sms(3);
        step=1;
    }

    if (PINF.1==0){
        //bacalagibuatdata();
        kirim_sms(1);
    }

    if (PINF.2==0){
        cek_modem();
    }
}

```

```

switch (step) {
case 0:
kontrol_posisi(0);break;
case 1:
    bacalagi();break;
case 2:
    lampu1=1;
    FirstError = ((-500*nilaibagi)-LastRotary) ;
    buka_brankas(-500);
    break;
case 3:
    lampu2=1;
    FirstError =(500*nilaibagi)-LastRotary ;
    buka_brankas(500);
    break;
case 4:
    lampu3=1;
    FirstError = ((0*nilaibagi)-LastRotary) ;
    buka_brankas(0);
    break;
case 5:
    lampu4=1;
    FirstError = (((data3-300)*nilaibagi)-LastRotary) ;
    buka_brankas((data3-300));
    break;
case 6:
    lampu5=1;
    FirstError = (((data4-200)*nilaibagi)-LastRotary) ;
    buka_brankas((data4-200));
    break;
case 7:
    final();
    break;
default:
};

```

```
    }  
    else{  
        lcd_gotoxy(0,0);lcd_puts("Error Sistem");  
        //delay_ms(10000);  
        m_stop();  
        kirim_sms(2);  
        delay_ms(600000);  
    }  
}  
}
```

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

DATASHEET

1. Datasheet Mikrokontroler ATmega128

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 133 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 128Kbytes of In-System Self-programmable Flash program memory
 - 4Kbytes EEPROM
 - 4Kbytes Internal SRAM
 - Write/Erase cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Up to 64Kbytes Optional External Memory Space
 - Programming Lock for Software Security
 - SPI Interface for In-System Programming
- QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Two 8-bit PWM Channels
 - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
 - Output Compare Modulator
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Dual Programmable Serial USARTs
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
 - Software Selectable Clock Frequency
 - ATmega103 Compatibility Mode Selected by a Fuse
 - Global Pull-up Disable
- I/O and Packages
 - 53 Programmable I/O Lines
 - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V ATmega128L
 - 4.5 - 5.5V ATmega128
- Speed Grades
 - 0 - 8MHz ATmega128L
 - 0 - 16MHz ATmega128



**8-bit Atmel
Microcontroller
with 128KBytes
In-System
Programmable
Flash**

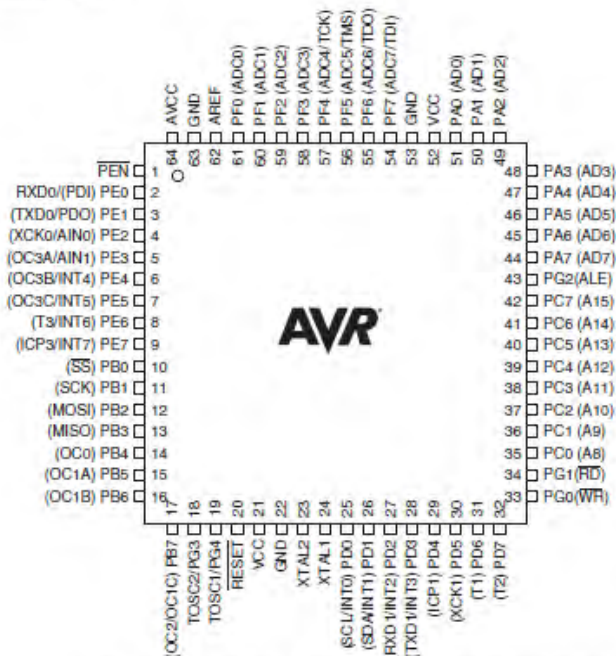
**ATmega128
ATmega128L**

Doc. 2467X-AVR0-00/1



Pin Configurations

Figure 1. Pinout ATmega128



Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the GFMMLF package should be soldered to ground.

Overview

The Atmel® AVR® ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Interrupts

This section describes the specifics of the interrupt handling as performed in Atmel® AVR® ATmega128. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 14.

Interrupt Vectors in ATmega128

Table 23. Reset and Interrupt Vectors

Vector No.	Program Address ¹⁾	Source	Interrupt Definition
1	\$0000 ¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$0002	INT0	External Interrupt Request 0
3	\$0004	INT1	External Interrupt Request 1
4	\$0006	INT2	External Interrupt Request 2
5	\$0008	INT3	External Interrupt Request 3
6	\$000A	INT4	External Interrupt Request 4
7	\$000C	INT5	External Interrupt Request 5
8	\$000E	INT6	External Interrupt Request 6
9	\$0010	INT7	External Interrupt Request 7
10	\$0012	TIMER2 COMP	Timer/Counter2 Compare Match
11	\$0014	TIMER2 OVF	Timer/Counter2 Overflow
12	\$0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	\$0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	\$001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	\$001C	TIMER1 OVF	Timer/Counter1 Overflow
16	\$001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	\$0020	TIMER0 OVF	Timer/Counter0 Overflow
18	\$0022	SPI, STC	SPI Serial Transfer Complete
19	\$0024	USART0, RX	USART0, Rx Complete
20	\$0026	USART0, UDRE	USART0 Data Register Empty
21	\$0028	USART0, TX	USART0, Tx Complete
22	\$002A	ADC	ADC Conversion Complete
23	\$002C	EE READY	EEPROM Ready
24	\$002E	ANALOG COMP	Analog Comparator
25	\$0030 ²⁾	TIMER1 COMPC	Timer/Counter1 Compare Match C
26	\$0032 ²⁾	TIMER3 CAPT	Timer/Counter3 Capture Event
27	\$0034 ²⁾	TIMER3 COMPA	Timer/Counter3 Compare Match A
28	\$0036 ²⁾	TIMER3 COMPB	Timer/Counter3 Compare Match B
29	\$0038 ²⁾	TIMER3 COMPC	Timer/Counter3 Compare Match C
30	\$003A ²⁾	TIMER3 OVF	Timer/Counter3 Overflow

Table 23. Reset and Interrupt Vectors

Vector No.	Program Address ⁽¹⁾	Source	Interrupt Definition
31	\$003C ⁽²⁾	USART1, RX	USART1, Rx Complete
32	\$003E ⁽²⁾	USART1, UDRE	USART1 Data Register Empty
33	\$0040 ⁽²⁾	USART1, TX	USART1, Tx Complete
34	\$0042 ⁽²⁾	I ² C	Two-wire Serial Interface
35	\$0044 ⁽²⁾	SPM READY	Store Program Memory Ready

- Notes:
1. When the BOOTRST fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 273.
 2. When the IVSEL bit in MCUCR is set, interrupt vectors will be moved to the start of the Boot Flash section. The address of each interrupt vector will then be address in this table added to the start address of the boot Flash section.
 3. The Interrupts on address \$0030 - \$0044 do not exist in ATmega103 compatibility mode.

Table 24 shows Reset and Interrupt vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt vectors are in the Boot section or vice versa.

Table 24. Reset and Interrupt Vectors Placement

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	\$0000	\$0002
1	1	\$0000	Boot Reset Address + \$0002
0	0	Boot Reset Address	\$0002
0	1	Boot Reset Address	Boot Reset Address + \$0002

- Note: The Boot Reset Address is shown in Table 112 on page 284. For the BOOTRST fuse "1" means unprogrammed while "0" means programmed.

Register Description for I/O Ports

Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	I/W	I/W	I/W	I/W	I/W	I/W	I/W	I/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	DDRA
Read/Write	I/W	I/W	I/W	I/W	I/W	I/W	I/W	I/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Input Pins Address – PINA

Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	I/W	I/W	I/W	I/W	I/W	I/W	I/W	I/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	DDRB
Read/Write	I/W	I/W	I/W	I/W	I/W	I/W	I/W	I/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	I/W	I/W	I/W	I/W	I/W	I/W	I/W	I/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	DDRC
Read/Write	I/W	I/W	I/W	I/W	I/W	I/W	I/W	I/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	NA	NA	NA	NA	NA	NA	NA	NA	

In Atmel® AVR®ATmega103 compatibility mode, DDRC and PINC Registers are initialized to being Push-Pull Zero Output. The port pins assumes their initial value, even if the clock is not running. Note that the DDRC and PINC Registers are available in ATmega103 compatibility mode, and should not be used for 100% back-ward compatibility.

Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	DDRD
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	NA	NA	NA	NA	NA	NA	NA	NA	

Port E Data Register – PORTE

Bit	7	6	5	4	3	2	1	0	
	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	PORTE
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port E Data Direction Register – DDRE

Bit	7	6	5	4	3	2	1	0	
	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0	DDRE
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port E Input Pins Address – PINE

Bit	7	6	5	4	3	2	1	0	
	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	PINE
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	NA	NA	NA	NA	NA	NA	NA	NA	

Port F Data Register – PORTF

Bit	7	6	5	4	3	2	1	0	
	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	PORTF
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port F Data Direction Register – DDRF

Bit	7	6	5	4	3	2	1	0	
	DDRF7	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0	DDRF
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port F Input Pins Address – PINF

Bit	7	6	5	4	3	2	1	0	
	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	PINF
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	NA	NA	NA	NA	NA	NA	NA	NA	

Note that PORTF and DDRF Registers are not available in Atmel® AVR®ATmega103 compatibility mode where Port F serves as digital input only.

Port G Data Register – PORTG

Bit	7	6	5	4	3	2	1	0	
	—	—	—	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	PORTG
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port G Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	—	—	—	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	DDRC
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Port G Input Pins Address – PING

Bit	7	6	5	4	3	2	1	0	
	—	—	—	PING4	PING3	PING2	PING1	PING0	PING
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	NA	NA	NA	NA	NA	

Note that PORTG, DDRC, and PING are not available in ATmega103 compatibility mode. In the ATmega103 compatibility mode Port G serves its alternate functions only (TOSC1, TOSC2, WR, RD and ALE).

Electrical Characteristics

Note: Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except RESET with respect to Ground	-0.5V to $V_{CC} + 0.5V$
Voltage on RESET with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 - 400.0mA

***NOTICE:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7V$ to $5.5V$ (unless otherwise noted)

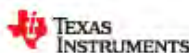
Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage except XTAL1 and RESET pins	$V_{CC} = 2.7V - 5.5V$	-0.5		$0.2 V_{CC}^{(1)}$	V
V_{IH}	Input High Voltage except XTAL1 and RESET pins	$V_{CC} = 2.7V - 5.5V$	$0.6 V_{CC}^{(2)}$		$V_{CC} + 0.5$	
V_{IL1}	Input Low Voltage XTAL1 pin	$V_{CC} = 2.7V - 5.5V$	-0.5		$0.1 V_{CC}^{(1)}$	
V_{IH1}	Input High Voltage XTAL1 pin	$V_{CC} = 2.7V - 5.5V$	$0.7 V_{CC}^{(2)}$		$V_{CC} + 0.5$	
V_{IL2}	Input Low Voltage RESET pin	$V_{CC} = 2.7V - 5.5V$	-0.5		$0.2 V_{CC}^{(1)}$	
V_{IH2}	Input High Voltage RESET pin	$V_{CC} = 2.7V - 5.5V$	$0.85 V_{CC}^{(2)}$		$V_{CC} + 0.5$	
V_{OL}	Output Low Voltage ⁽³⁾ (Ports A,B,C,D, E, F, G)	$I_{OL} = 20mA, V_{CC} = 5V$ $I_{OL} = 10mA, V_{CC} = 3V$			0.7 0.5	V
V_{OH}	Output High Voltage ⁽³⁾ (Ports A,B,C,D, E, F, G)	$I_{OH} = -20mA, V_{CC} = 5V$ $I_{OH} = -10mA, V_{CC} = 3V$	4.2 2.2			V
I_{IL}	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$, pin low (absolute value)			1.0	μA
I_{IH}	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$, pin high (absolute value)			1.0	
R_{RES1}	Reset Pull-up Resistor		30		85	k Ω
R_{PEN}	PEN Pull-up Resistor		30		60	
R_{PU}	I/O Pin Pull-up Resistor		20		50	

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7\text{V}$ to 5.5V (unless otherwise noted) (Continued)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{CC}	Power Supply Current	Active 4MHz, $V_{CC} = 3\text{V}$ (ATmega128L)		5	5.5	mA
		Active 8MHz, $V_{CC} = 5\text{V}$ (ATmega128)		17	19	
		Idle 4MHz, $V_{CC} = 3\text{V}$ (ATmega128L)		2	2.5	
		Idle 8MHz, $V_{CC} = 5\text{V}$ (ATmega128)		8	11	
	Power-down mode	WDT enabled, $V_{CC} = 3\text{V}$		< 15	25	μA
		WDT disabled, $V_{CC} = 3\text{V}$		< 5	10	
V_{AGC}	Analog Comparator Input Offset Voltage	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$			40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$	-50		50	nA
t_{ACPD}	Analog Comparator Propagation Delay	$V_{CC} = 2.7\text{V}$ $V_{CC} = 5.0\text{V}$		700 500		ns

- Notes:
- "Max" means the highest value where the pin is guaranteed to be read as low
 - "Min" means the lowest value where the pin is guaranteed to be read as high
 - Although each I/O port can sink more than the test conditions (20mA at $V_{CC} = 5\text{V}$, 10mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
 TQFP and QFNMLF Package:
 1) The sum of all IOL, for all ports, should not exceed 400mA.
 2) The sum of all IOL, for ports A0 - A7, G2, C3 - C7 should not exceed 100mA.
 3) The sum of all IOL, for ports C0 - C2, G0 - G1, D0 - D7, XTAL2 should not exceed 100mA.
 4) The sum of all IOL, for ports B0 - B7, G3 - G4, E0 - E7 should not exceed 100mA.
 5) The sum of all IOL, for ports F0 - F7, should not exceed 100mA.
 If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
 - Although each I/O port can source more than the test conditions (20mA at $V_{CC} = 5\text{V}$, 10mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
 TQFP and QFNMLF Package:
 1) The sum of all IOH, for all ports, should not exceed 400mA.
 2) The sum of all IOH, for ports A0 - A7, G2, C3 - C7 should not exceed 100mA.
 3) The sum of all IOH, for ports C0 - C2, G0 - G1, D0 - D7, XTAL2 should not exceed 100mA.
 4) The sum of all IOH, for ports B0 - B7, G3 - G4, E0 - E7 should not exceed 100mA.
 5) The sum of all IOH, for ports F0 - F7, should not exceed 100mA.
 If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

2. Datasheet LM2596



LM2596

www.ti.com

SNVS124C – NOVEMBER 1996 – REVISED APRIL 2013

LM2596 SIMPLE SWITCHER® Power Converter 150 kHz 3A Step-Down Voltage Regulator

Check for Samples: LM2596

FEATURES

- 3.3V, 5V, 12V, and Adjustable Output Versions
- Adjustable Version Output Voltage Range, 1.2V to 37V $\pm 4\%$ Max Over Line and Load Conditions
- Available in TO-220 and TO-263 Packages
- Ensured 3A Output Load Current
- Input Voltage Range Up to 40V
- Requires Only 4 External Components
- Excellent Line and Load Regulation Specifications
- 150 kHz Fixed Frequency Internal Oscillator
- TTL Shutdown Capability
- Low Power Standby Mode, I_q Typically 80 μ A
- High Efficiency
- Uses Readily Available Standard Inductors
- Thermal Shutdown and Current Limit Protection

APPLICATIONS

- Simple High-Efficiency Step-Down (Buck) Regulator
- On-Card Switching Regulators
- Positive to Negative Converter

DESCRIPTION

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5V, 12V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed-frequency oscillator.

The LM2596 series operates at a switching frequency of 150 kHz thus allowing smaller sized filter components than what would be needed with lower frequency switching regulators. Available in a standard 5-lead TO-220 package with several different lead bend options, and a 5-lead TO-263 surface mount package.

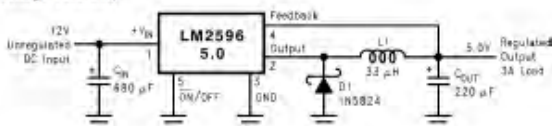
A standard series of inductors are available from several different manufacturers optimized for use with the LM2596 series. This feature greatly simplifies the design of switch-mode power supplies.

Other features include a ensured $\pm 4\%$ tolerance on output voltage under specified input voltage and output load conditions, and $\pm 15\%$ on the oscillator frequency. External shutdown is included, featuring typically 80 μ A standby current. Self protection features include a two stage frequency reducing current limit for the output switch and an over temperature shutdown for complete protection under fault conditions. ⁽¹⁾

(1) \dagger Patent Number 5,382,916.

Typical Application

(Fixed Output Voltage Versions)



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

SIMPLE SWITCHER is a registered trademark of Texas Instruments.

All other trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1996–2013, Texas Instruments Incorporated

Connection Diagrams

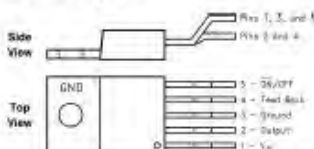


Figure 1. 5-Lead Bent and Staggered Leads, Through Hole TO-220 (T) Package
See Package Number NDH0005D

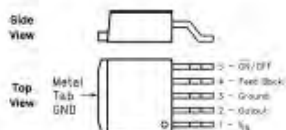


Figure 2. 5-Lead DDPak/TO-263 (S) Package
See Package Number KTT0005B



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

Absolute Maximum Ratings (1)(2)

Maximum Supply Voltage	45V
ON/OFF Pin Input Voltage	-0.3 s V s +25V
Feedback Pin Voltage	-0.3 s V s +25V
Output Voltage to Ground (Steady State)	-1V
Power Dissipation	Internally limited
Storage Temperature Range	-65°C to +150°C
ESD Susceptibility	
Human Body Model (3)	2 kV
Lead Temperature	
DDPAK/TO-263 Package	
Vapor Phase (60 sec.)	+215°C
Infrared (10 sec.)	+245°C
TO-220 Package (Soldering, 10 sec.)	+260°C
Maximum Junction Temperature	+150°C

- (1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not ensure specific performance limits. For assured specifications and test conditions, see the Electrical Characteristics.
- (2) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
- (3) The human body model is a 100 pF capacitor discharged through a 1.5k resistor into each pin.

Operating Conditions

Temperature Range	-40°C s T _J s +125°C
Supply Voltage	4.5V to 40V

Table 1. LM2596 Fixed Voltage Quick Design Component Selection Table (continued)

Conditions			Inductor		Output Capacitor			
Output Voltage (V)	Load Current (A)	Max Input Voltage (V)	Inductance (μH)	Inductor (#)	Through Hole Electrolytic		Surface Mount Tantalum	
5	3	8	22	L41	470/25	560/16	220/10	330/10
		10	22	L41	560/25	560/25	220/10	330/10
		15	33	L40	330/35	330/35	220/10	330/10
		40	47	L39	330/35	270/35	220/10	330/10
		9	22	L33	470/25	560/16	220/10	330/10
	2	20	68	L38	180/35	180/35	100/10	270/10
		40	68	L38	180/35	180/35	100/10	270/10
		15	22	L41	470/25	470/25	100/16	180/16
		18	33	L40	330/25	330/25	100/16	180/16
		30	68	L44	180/25	180/25	100/16	120/20
12	3	40	68	L44	180/35	180/35	100/16	120/20
		15	33	L32	330/25	330/25	100/16	180/16
		20	68	L38	180/25	180/25	100/16	120/20
		40	150	L42	82/25	82/25	68/20	68/25
	2	20	68	L38	180/25	180/25	100/16	120/20
		40	150	L42	82/25	82/25	68/20	68/25
		15	33	L32	330/25	330/25	100/16	180/16
		20	68	L38	180/25	180/25	100/16	120/20

LM2596 Series Buck Regulator Design Procedure (Adjustable Output)

PROCEDURE (Adjustable Output Voltage Version)	EXAMPLE (Adjustable Output Voltage Version)
<p>Given:</p> <p>V_{OUT} = Regulated Output Voltage</p> <p>$V_{IN(max)}$ = Maximum Input Voltage</p> <p>$I_{LOAD(max)}$ = Maximum Load Current</p> <p>F = Switching Frequency (Fixed at a nominal 150 kHz)</p> <p>1. Programming Output Voltage (Selecting R_1 and R_2 as shown in Figure 20.)</p> <p>Use the following formula to select the appropriate resistor values.</p> $V_{OUT} = V_{REF} \left(1 + \frac{R_2}{R_1} \right) \text{ where } V_{REF} = 1.23V \quad (1)$ <p>Select a value for R_1 between 240Ω and 1.5 kΩ. The lower resistor values minimize noise pickup in the sensitive feedback pin. (For the lowest temperature coefficient and the best stability with time, use 1% metal film resistors.)</p> $R_2 = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right) \quad (2)$	<p>Given:</p> <p>$V_{OUT} = 20V$</p> <p>$V_{IN(max)} = 28V$</p> <p>$I_{LOAD(max)} = 3A$</p> <p>F = Switching Frequency (Fixed at a nominal 150 kHz)</p> <p>1. Programming Output Voltage (Selecting R_1 and R_2 as shown in Figure 20.)</p> <p>Select R_1 to be 1 kΩ, 1%. Solve for R_2.</p> $R_2 = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right) = 1k \left(\frac{20V}{1.23V} - 1 \right) \quad (3)$ <p>$R_2 = 1k (16.26 - 1) = 15.26k$, closest 1% value is 15.4 kΩ.</p> <p>$R_2 = 15.4 kΩ$</p>

3. Datasheet Modul GSM SIM 900A



2 SIM900 Overview

Designed for global market, SIM900 is a quad-band GSM/GPRS engine that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM900 features GPRS multi-slot class 10/ class 8 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

With a tiny configuration of 24mm x 24mm x 3mm, SIM900 can meet almost all the space requirements in your applications, such as M2M, smart phone, PDA and other mobile devices.

The physical interface to the mobile application is a 68-pin SMT pad, which provides all hardware interfaces between the module and customers' boards.

- The keypad and SPI display interface will give you the flexibility to develop customized applications.
- Serial port and Debug port can help you easily develop your applications.
- One audio channel includes a microphone input and a speaker output.

The SIM900 is designed with power saving technique so that the current consumption is as low as 1.5mA in SLEEP mode.

The SIM900 is integrated with the TCP/IP protocol, extended TCP/IP AT commands are developed for customers to use the TCP/IP protocol easily, which is very useful for those data transfer applications.

2.1 SIM900 Key Features

Table 3: SIM900 key features:

Feature	Implementation
Power supply	Single supply voltage 3.4V – 4.5V
Power saving	Typical power consumption in SLEEP mode is 1.5mA (BS-PA-MFRMS=f)
Frequency Bands	<ul style="list-style-type: none">● SIM900 quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. The SIM900 can search the 4 frequency bands automatically. The frequency bands also can be set by AT command.● Compliant to GSM Phase 2/2+
GSM class	Small MS
Transmitting power	<ul style="list-style-type: none">● Class 4 (2W) at GSM 850 and EGSM 900● Class 1 (1W) at DCS 1800 and PCS 1900
GPRS connectivity	<ul style="list-style-type: none">● GPRS multi-slot class 10 (default)● GPRS multi-slot class 8 (option)● GPRS mobile station class B
Temperature range	<ul style="list-style-type: none">● Normal operation: -30°C to +60°C

SIM900 Hardware Design

	<ul style="list-style-type: none"> ● Restricted operation: -40°C to -30°C and +80 °C to +35°C⁽¹⁾ ● Storage temperature -45°C to +90°C
DATA GPRS	<ul style="list-style-type: none"> ● GPRS data downlink transfer: max. 83.6 kbps ● GPRS data uplink transfer: max. 42.8 kbps ● Coding scheme: CS-1, CS-2, CS-3 and CS-4 ● SIM900 supports the protocols PAP (Password Authentication Protocol) usually used for PPP connections. ● The SIM900 integrates the TCP/IP protocol. ● Support Packet Switched Broadcast Control Channel (PBCCH)
CSD	<ul style="list-style-type: none"> ● CSD transmission rates: 2.4, 4.8, 9.6, 14.4 kbps, non-transparent ● Unstructured Supplementary Services Data (USSD) support
SMS	<ul style="list-style-type: none"> ● MT, MO, CB, Text and PDU mode ● SMS storage: SIM card
FAX	Group 3 Class 1
SIM interface	Support SIM card: 1.8V, 3V
External antenna	Antenna pad
Audio features	Speech codec modes: <ul style="list-style-type: none"> ● Half Rate (ETS 06.20) ● Full Rate (ETS 06.10) ● Enhanced Full Rate (ETS 06.10 / 06.60 / 06.80) ● Adaptive multi rate (AMR) ● Echo Cancellation ● Noise Suppression
Serial port and Debug port	Serial Port: <ul style="list-style-type: none"> ● 8-wire modem interface with status and control lines, unbalanced, asynchronous. ● 1.2kbps to 11.52kbps. ● Serial Port can be used for AT commands or data stream. ● Supports RTS/CTS hardware handshake and software ON/OFF flow control. ● Multiplex ability according to GSM 07.10 Multiplexer Protocol. ● Autobauding supports baud rate from 1200 bps to 115200bps. Debug port: <ul style="list-style-type: none"> ● 2-wire null modem interface DBG_TXD and DBG_RXD ● Can be used for debugging and upgrading firmware.
Phonebook management	Support phonebook types: SM, FD, LD, RC, ON, MC.
SIM Application Toolkit	Support SAT class 3, GSM 11.14 Release 99
Real time clock	Implemented
Timer function	Programmable via AT command
Physical characteristics	Size: 24mm x 24mm x 3mm Weight: 3.4g
Firmware upgrade	Firmware upgrade by debug port.

SIM900 Hardware Design

SDA	I/O	I ² C bus data		open
KBR0~KBR4	O			If unused keep these pins open
KBC0~KBC4	I	Keypad interface		Pull up to VDD_EXT, if unused keep pins open

Serial port

PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
RXD	I	Receive data	$V_{ILmax}=0.15 \cdot VDD_EXT$	DTR Pin has been pulled up to VDD_EXT. If unused keep pins open
TXD	O	Transmit data	$V_{IHmin}=0.85 \cdot VDD_EXT$	
RTS	I	Request to send	$V_{ILmin}=0V$	
CTS	O	Clear to send	$V_{IHmax}=VDD_EXT$	
RI	O	Ring indicator	$VOHmin=VDD_EXT-0.1V$	
DSR	O	Data Set Ready	$VOLmax=0.1V$	
DCD	O	Data carry detect	$VOHmax=VDD_EXT$	
DTR	I	Data terminal Ready	$VOLmin=0V$	

Debug interface

PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
DBG_TXD	O	Serial interface for debugging and firmware upgrade	$V_{ILmax}=0.15 \cdot VDD_EXT$	If unused keep pins open
			$V_{IHmin}=0.85 \cdot VDD_EXT$	
			$V_{ILmin}=0V$	
			$V_{IHmax}=VDD_EXT$	
			$VOHmin=VDD_EXT-0.1V$	
			$VOLmax=0.1V$	
			$VOHmax=VDD_EXT$	
			$VOLmin=0V$	

SIM interface

PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
SIM_VDD	O	Voltage supply for SIM card	The voltage can be select by software automatically either 1.8V or 3V	
SIM_DATA	I/O	SIM data output	$V_{ILmax}=0.15 \cdot SIM_VDD$	All signals of SIM interface are protected against ESD with a TVS diode array. Maximum cable length 200mm from the module
SIM_CLK	O	SIM clock	$V_{IHmin}=0.85 \cdot SIM_VDD$	
		SIM reset	$V_{ILmin}=0V$	
			$V_{IHmax}=SIM_VDD$	
			$VOHmin=SIM_VDD-0.1V$	
			$VOLmax=0.1V$	
			$VOHmax=SIM_VDD$	
			$VOLmin=0V$	

• Li-battery backup

Coin-type Rechargeable Capacitor such as XH414H-IV01E from Seiko can be used.

Typical charge curves for each cell type are shown in following figures. Note that the rechargeable Lithium type coin cells generally come pre-charged from the vendor.

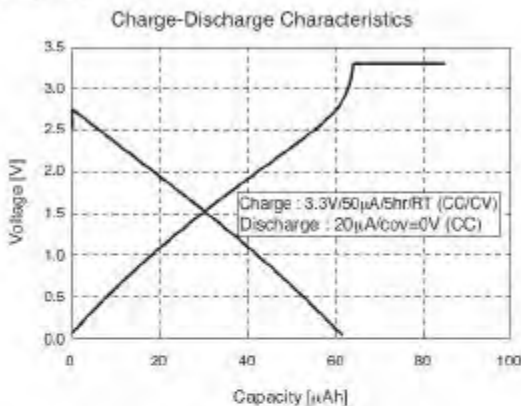


Figure 19: Seiko XH414H-IV01E Charge Characteristic

3.8 Serial Interfaces

Table 7: Pin definition of the serial interfaces

	Name	Pin	Function
Serial port	DTR	3	Data terminal ready
	RI	4	Ring indicator
	DCD	5	Data carrier detection
	DSR	6	Data set ready
	CTS	7	Clear to send
	RTS	8	Request to send
	TXD	9	Transmit data
	RXD	10	Receive data
Debug port	DBG_RXD	28	Receive data
	DBG_TXD	27	Transmit data

SIM900 provides two unbalanced asynchronous serial ports. One is the serial port and the other is the debug port.

The GSM module is designed as a DCE (Data Communication Equipment), following the traditional DCE-DTE (Data Terminal Equipment) connection. The module and the client (DTE) are connected through the following signal (as following figure shows). Autobauding supports baud rate from 1200bps to 115200bps.

Serial port

- TXD: Send data to the RXD signal line of the DTE
- RXD: Receive data from the TXD signal line of the DTE

Debug port

- DBG_TXD: Send data to the RXD signal line of the DTE
- DBG_RXD: Receive data from the TXD signal line of the DTE

The logic levels are described in following table.

Table 8: Logic levels of the serial port and debug port

Parameter	Min	Max	Unit
V_{IL}	0	$0.15 \times VDD_EXT$	V
V_{IH}	$0.85 \times VDD_EXT$	VDD_EXT	V
V_{OL}	0	0.1	V
V_{OH}	$VDD_EXT - 0.1$	VDD_EXT	V

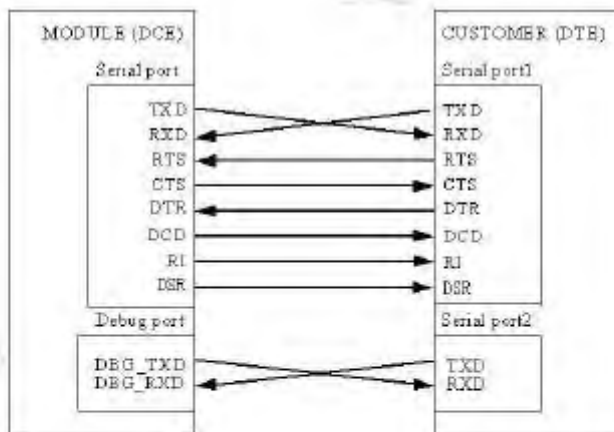


Figure 20: Connection of the serial interfaces

6.2 Top and Bottom View of the SIM900



Figure 39: Top view of the SIM900

6.3 PIN Assignment of SIM900

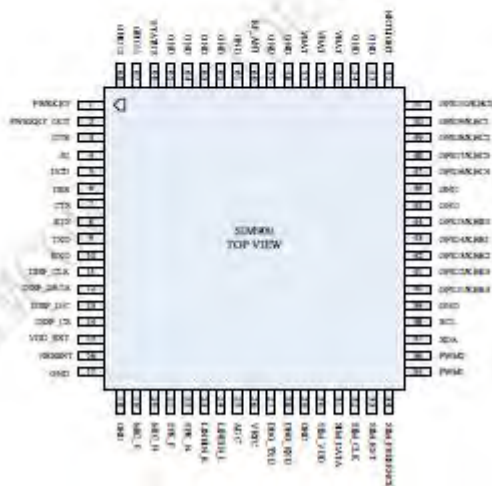


Figure 40: SIM900 pin out diagram (Top View)

Table 31: PIN assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	PWRKEY	35	PMW1
2	PWRKEY_OUT	36	PWM2
3	DTR	37	SDA
4	RI	38	SCL
5	DCD	39	GND
6	DSR	40	GPIO1/KBR4
7	CTS	41	GPIO2/KBR3
8	RTS	42	GPIO3/KBR2
9	TXD	43	GPIO4/KBR1
10	RXD	44	GPIO5/KBR0
11	DISP_CLK	45	GND
12	DISP_DATA	46	GND
13	DISP_D/C	47	GPIO6/KBC4
14	DISP_CS	48	GPIO7/KBC3
15	VDD_EXT	49	GPIO8/KBC2
16	NRESET	50	GPIO9/KBC1
17	GND	51	GPIO10/KBC0
18	GND	52	NETLIGHT
19	MIC_P	53	GND
20	MIC_N	54	GND
21	SPK_P	55	VBAT
22	SPK_N	56	VBAT
23	LINEN_R	57	VBAT
24	LINEN_L	58	GND
25	ADC	59	GND
26	VRTC	60	RF_ANT
27	DBG_TXD	61	GND
28	DBG_RXD	62	GND
29	GND	63	GND
30	SIM_VDD	64	GND
31	SIM_DATA	65	GND
32	SIM_CLK	66	STATUS
33	SIM_RST	67	GPIO11
34	SIM_PRESENCE	68	GPIO12

4 AT Commands According to GSM07.05

The GSM 07.05 commands are for performing SMS and CBS related operations. SIM900 supports both Text and PDU modes.

4.1 Overview of AT Commands According to GSM07.05

Command	Description
AT+CMGD	DELETE SMS MESSAGE
AT+CMGF	SELECT SMS MESSAGE FORMAT
AT+CMGL	LIST SMS MESSAGES FROM PREFERRED STORE
AT+CMGR	READ SMS MESSAGE
AT+CMGS	SEND SMS MESSAGE
AT+CMGW	WRITE SMS MESSAGE TO MEMORY
AT+CMSS	SEND SMS MESSAGE FROM STORAGE
AT+CNMI	NEW SMS MESSAGE INDICATIONS
AT+CPMS	PREFERRED SMS MESSAGE STORAGE
AT+CRFS	RESTORE SMS SETTINGS
AT+CSAS	SAVE SMS SETTINGS
AT+CSCA	SMS SERVICE CENTER ADDRESS
AT+CSCB	SELECT CELL BROADCAST SMS MESSAGES
AT+CSDB	SHOW SMS TEXT MODE PARAMETERS
AT+CSMP	SET SMS TEXT MODE PARAMETERS
AT+CSMS	SELECT MESSAGE SERVICE

4.2 Detailed Descriptions of AT Commands According to GSM07.05

4.2.1 AT+CMGD Delete SMS Message

AT+CMGD Delete SMS Message	
Test Command	Response
AT+CMGD=?	+CMGD: (list of supported <index>),(list of supported <delflag>)
	OK
	Parameters
	See Write Command
Write Command	Response
AT+CMGD=<index>[,<delflag>]	TA deletes message from preferred message storage <meml> location <index>.
	OK
	ERROR

13 AT Commands Sample

13.1 Profile Commands

Demonstration	Syntax	Expect Result
The AT Command interpreter actively responds to input	AT	OK
Display the product name and the product release information.	ATI	SIM900 R11.0
Display product identification information: the manufacturer, the product name and the product revision information.	AT+GSV	SIMCOM_Ltd SIMCOM_SIM900 Revision:1137B01SIM900M32_ST OK
Display current configuration, a list of the current active profile parameters.	AT+V	[A complete listing of the active profile] OK
Reporting of mobile equipment errors. The default CME error reporting setting is disabled.	AT+CMEE=?	+CMEE:(0-2)
Switch to verbose mode	AT+CMEE?	OK +CMEE: 1
	AT+CSCS=?	OK +CSCS: ("IRA","GSM","UCS2","HEX","PCP","PCDN","8859-1")
	AT+CSCS="TEST"	OK

Displays a string explaining the error in more details.	AT+CMEEE=2	ERROR
Store the current configuration in nonvolatile memory. When the board is reset, the configuration changes from the last session are loaded.	AT+CSCS="TEST"	OK
		+CME ERROR: operation not allowed
	ATE0&W	OK
	AT	[No echo]
		OK
	[Reset the board]	
	AT	[No echo]
		OK
	ATE1&W	[No echo]
		OK
	AT	[Echo on]
		OK
Set the ME to minimum functionality	AT+IPR?	+IPR:0
		OK
	AT+CFUN=0	OK
	AT+IPR=115200	+CPIN: NOT READY
		OK
	AT+IPR?	+IPR:115200
	AT+CFUN=0	OK
		+CPIN: NOT READY

ME has entered full functionality mode.	AT+CFUN?	+CFUN:1
		OK

13.2 SIM Commands

Demonstration	Syntax	Expect Result
List available phonebooks, and select the SIM phonebook.	AT+CPBS=?	+CPBS: ("MC","RC","DC","LD","LA","ME","SM","FD", "ON","BN","SD","VM","EN")
		OK

<p>Unsolicited notification of the SMS arriving</p> <p>Read SMS message that has just arrived.</p> <p>Note: the number should be the same as that given in the +CMTI notification.</p>	<p>AT+CMGS="+8613918186089" >This is a test <Ctrl+Z></p>	<p>+CMGS:34</p> <p>OK</p> <p>+CMTI:"SM",1</p>
<p>Reading the message again and change the status to "READ" from "UNREAD"</p>	<p>AT+CMGR=1</p> <p>AT+CMGR=1</p>	<p>+CMGR: "REC UNREAD", "+8613918186089", "", "02/01/30,20:40:31+00" This is a test</p> <p>OK</p> <p>+CMGR: "REC READ", "+8613918186089", "", "02/01/30,20:40:31+00" This is a test</p>
<p>Send another SMS to myself.</p>	<p>AT+CMGS="+8613918186089" >Test again<Ctrl+Z></p>	<p>+CMGS:35</p> <p>OK</p>
<p>Unsolicited notification of the SMS arriving</p> <p>List all SMS messages.</p> <p>Note:"ALL" must be in uppercase.</p>	<p>AT+CMGL="ALL"</p>	<p>+CMTI:"SM",2</p> <p>+CMGL: 1, "REC READ", "+8613918186089", "", "02/01/30,20:40:31+00" This is a test</p> <p>+CMGL: 2, "REC UNREAD", "", "+8613918186089", "", "02/01/30,20:45:12+00" Test again</p> <p>OK</p>
<p>Delete an SMS message.</p> <p>List all SMS messages to show message has been deleted.</p>	<p>AT+CMGD=1</p> <p>AT+CMGL="ALL"</p>	<p>OK</p> <p>+CMGL: 2, "REC READ", "+8613918186089", "", "02/01/30,20:45:12+00" Test again</p>

4. Datasheet Limit Switch

SS

Subminiature Basic Switch

Subminiature Basic Switch Offers High Reliability and Security

- The OMRON's best-selling micro switches of a wide variety from 0.1A to 10.1A.
- A variety of models are available, with operating force ranging from low to high.
- Two split springs ensure a high stability and durability of 30,000,000 operations.

RoHS Compliant





Model Number Legend

- SS- [1] [2] [3] [4] [5] [6]
- 1. Ratings**
 - 10 : 250 VAC 10.1A
 - 5 : 125 VAC 5 A
 - 01 : 30 VDC 0.1A
 - 2. Actuator**
 - None : Pin plunger
 - GL : Hinge lever
 - GL111 : Long hinge lever
 - GL13 : Simulated roller lever
 - GL2 : Hinge roller lever
 - GL02 : Hinge roller lever (Roller material: Stainless) heat resistant
 - 3. Maximum Operating Force (OP)**
 - None : 1.47 N (150 g)
 - F : 0.49 N (50 g) (0.1 A, 5 A)
 - E : 0.25 N (25 g) (0.1 A)




Note: These values are for the pin plunger models.
 - 4. Contact Form**
 - None : SPDT
 - 2 : SPST-NC
 - 3 : SPST-NO
 - 5. Terminals**
 - None : Solder terminals
 - T : Quick-connect terminals (#110)
 - D : PCB terminals
 - 6. Heat resistance**
 - None : Standard (85°C)
 - T : Heat-resistant (120°C)

List of Models

Standard Models

Actuator	Terminals	Contact Form	Maximum Operating Force (OP)	10.1 A	5 A	0.1 A
Pin plunger 	Solder terminals	SPDT	1.47 N (150 g)	SS-10	SS-5	SS-01
		SPDT-NC		SS-10-2	SS-5-2	SS-01-2
		SPDT-NO		SS-10-3	SS-5-3	SS-01-3
	Quick-connect terminals (#110)	SPDT		SS-10T	SS-5T	SS-01T
		SPDT-NC		SS-10-2T	SS-5-2T	SS-01-2T
		SPDT-NO		SS-10-3T	SS-5-3T	SS-01-3T
	PCB terminals	SPDT		SS-10S	SS-5S	SS-01S
		SPDT-NC		SS-10-2S	SS-5-2S	SS-01-2S
		SPDT-NO		SS-10-3S	SS-5-3S	SS-01-3S
	Solder terminals	SPDT	0.49 N (50 g)	-	SS-5-F	SS-01-F
		SPDT-NC		-	SS-5-F-2	SS-01-F-2
		SPDT-NO		-	SS-5-F-3	SS-01-F-3
	Quick-connect terminals (#110)	SPDT		-	SS-5-F-T	SS-01-F-T
		SPDT-NC		-	SS-5-F-2T	SS-01-F-2T
		SPDT-NO		-	SS-5-F-3T	SS-01-F-3T
	PCB terminals	SPDT		-	SS-5-F-S	SS-01-F-S
		SPDT-NC		-	SS-5-F-2S	SS-01-F-2S
		SPDT-NO		-	SS-5-F-3S	SS-01-F-3S
Pin plunger 	Solder terminals	SPDT	0.25 N (25 g)	-	-	SS-01-E-2
		SPDT-NC		-	-	SS-01-E-2-2
		SPDT-NO		-	-	SS-01-E-2-3
	Quick-connect terminals (#110)	SPDT		-	-	SS-01-E-T
		SPDT-NC		-	-	SS-01-E-2T
		SPDT-NO		-	-	SS-01-E-3T
	PCB terminals	SPDT		-	-	SS-01-E-S
		SPDT-NC		-	-	SS-01-E-2S
		SPDT-NO		-	-	SS-01-E-3S

Separator (Sold Separately), Terminal Connector (Sold Separately) ⇨ Refer to "Basic Switch Common Accessories"

Actuator	Terminals	Contact Form	Maximum Operating Force (gF)	Ratings		
				10.1 A	5 A	0.1 A
 Hinge lever	Solder terminals	SPDT	0.29 N (25 gF)	00-100L	00-100L	00-010L
		SPDT-A20		00-100L-2	00-100L-2	00-010L-2
		SPDT-A30		00-100L-3	00-100L-3	00-010L-3
	Quick-connect terminals (W18)	SPDT		00-100L-T	00-100L-T	00-010L-T
		SPDT-A20		00-100L-T-2	00-100L-T-2	00-010L-T-2
		SPDT-A30		00-100L-T-3	00-100L-T-3	00-010L-T-3
	PCB terminals	SPDT		00-100L-B	00-100L-B	00-010L-B
		SPDT-A20		00-100L-B-2	00-100L-B-2	00-010L-B-2
		SPDT-A30		00-100L-B-3	00-100L-B-3	00-010L-B-3
	Solder terminals	SPDT	0.16 N (16 gF)	-	00-100L-F	00-010L-F
		SPDT-A20		-	00-100L-F-2	00-010L-F-2
		SPDT-A30		-	00-100L-F-3	00-010L-F-3
	Quick-connect terminals (W18)	SPDT		-	00-100L-F-T	00-010L-F-T
		SPDT-A20		-	00-100L-F-T-2	00-010L-F-T-2
		SPDT-A30		-	00-100L-F-T-3	00-010L-F-T-3
	PCB terminals	SPDT		-	00-100L-F-B	00-010L-F-B
		SPDT-A20		-	00-100L-F-B-2	00-010L-F-B-2
		SPDT-A30		-	00-100L-F-B-3	00-010L-F-B-3
	Solder terminals	SPDT	0.06 N (6 gF)	-	-	00-010L-E
		SPDT-A20		-	-	00-010L-E-2
		SPDT-A30		-	-	00-010L-E-3
	Quick-connect terminals (W18)	SPDT		-	-	00-010L-E-T
		SPDT-A20		-	-	00-010L-E-T-2
		SPDT-A30		-	-	00-010L-E-T-3
	PCB terminals	SPDT		-	-	00-010L-E-B
		SPDT-A20		-	-	00-010L-E-B-2
		SPDT-A30		-	-	00-010L-E-B-3
 Long hinge lever	Solder terminals	SPDT	0.39 N (35 gF)	00-100L111	00-100L111	00-010L111
		SPDT-A20		00-100L111-2	00-100L111-2	00-010L111-2
		SPDT-A30		00-100L111-3	00-100L111-3	00-010L111-3
	Quick-connect terminals (W18)	SPDT		00-100L111-T	00-100L111-T	00-010L111-T
		SPDT-A20		00-100L111-T-2	00-100L111-T-2	00-010L111-T-2
		SPDT-A30		00-100L111-T-3	00-100L111-T-3	00-010L111-T-3
	PCB terminals	SPDT		00-100L111-B	00-100L111-B	00-010L111-B
		SPDT-A20		00-100L111-B-2	00-100L111-B-2	00-010L111-B-2
		SPDT-A30		00-100L111-B-3	00-100L111-B-3	00-010L111-B-3
	Solder terminals	SPDT	0.12 N (12 gF)	-	00-100L111-F	00-010L111-F
		SPDT-A20		-	00-100L111-F-2	00-010L111-F-2
		SPDT-A30		-	00-100L111-F-3	00-010L111-F-3
	Quick-connect terminals (W18)	SPDT		-	00-100L111-F-T	00-010L111-F-T
		SPDT-A20		-	00-100L111-F-T-2	00-010L111-F-T-2
		SPDT-A30		-	00-100L111-F-T-3	00-010L111-F-T-3
	PCB terminals	SPDT		-	00-100L111-F-B	00-010L111-F-B
		SPDT-A20		-	00-100L111-F-B-2	00-010L111-F-B-2
		SPDT-A30		-	00-100L111-F-B-3	00-010L111-F-B-3
	Solder terminals	SPDT	0.06 N (6 gF)	-	-	00-010L111-E
		SPDT-A20		-	-	00-010L111-E-2
		SPDT-A30		-	-	00-010L111-E-3
	Quick-connect terminals (W18)	SPDT		-	-	00-010L111-E-T
		SPDT-A20		-	-	00-010L111-E-T-2
		SPDT-A30		-	-	00-010L111-E-T-3
	PCB terminals	SPDT		-	-	00-010L111-E-B
		SPDT-A20		-	-	00-010L111-E-B-2
		SPDT-A30		-	-	00-010L111-E-B-3
 Simulated roller lever	Solder terminals	SPDT	0.49 N (50 gF)	00-100L13	00-100L13	00-010L13
		SPDT-A20		00-100L13-2	00-100L13-2	00-010L13-2
		SPDT-A30		00-100L13-3	00-100L13-3	00-010L13-3
	Quick-connect terminals (W18)	SPDT		00-100L13-T	00-100L13-T	00-010L13-T
		SPDT-A20		00-100L13-T-2	00-100L13-T-2	00-010L13-T-2
		SPDT-A30		00-100L13-T-3	00-100L13-T-3	00-010L13-T-3
	PCB terminals	SPDT		00-100L13-B	00-100L13-B	00-010L13-B
		SPDT-A20		00-100L13-B-2	00-100L13-B-2	00-010L13-B-2
		SPDT-A30		00-100L13-B-3	00-100L13-B-3	00-010L13-B-3
	Solder terminals	SPDT	0.16 N (16 gF)	-	00-100L13-F	00-010L13-F
		SPDT-A20		-	00-100L13-F-2	00-010L13-F-2
		SPDT-A30		-	00-100L13-F-3	00-010L13-F-3
	Quick-connect terminals (W18)	SPDT		-	00-100L13-F-T	00-010L13-F-T
		SPDT-A20		-	00-100L13-F-T-2	00-010L13-F-T-2
		SPDT-A30		-	00-100L13-F-T-3	00-010L13-F-T-3
	PCB terminals	SPDT		-	00-100L13-F-B	00-010L13-F-B
		SPDT-A20		-	00-100L13-F-B-2	00-010L13-F-B-2
		SPDT-A30		-	00-100L13-F-B-3	00-010L13-F-B-3
	Solder terminals	SPDT	0.06 N (6 gF)	-	-	00-010L13-E
		SPDT-A20		-	-	00-010L13-E-2
		SPDT-A30		-	-	00-010L13-E-3
	Quick-connect terminals (W18)	SPDT		-	-	00-010L13-E-T
		SPDT-A20		-	-	00-010L13-E-T-2
		SPDT-A30		-	-	00-010L13-E-T-3
	PCB terminals	SPDT		-	-	00-010L13-E-B
		SPDT-A20		-	-	00-010L13-E-B-2
		SPDT-A30		-	-	00-010L13-E-B-3

Separator (Sold Separately), Terminal Connector (Sold Separately) * Refers to "Basic Switch Common Accessories"

Ratings

Model	Rated voltage	Resistive load
GS-10 models	250 VAC	10.1 A
GS-5 models	125 VAC	5 A
	250 VAC	3 A
GS-01 models	125 VAC	0.1 A
	50 VDC	0.1 A

Note: The above rating values apply under the following test conditions:

- (1) Ambient temperature: $20 \pm 2^\circ\text{C}$
(2) Ambient humidity: $65 \pm 5\%$
(3) Operating frequency: 30 operations/min

Approved Safety Standards

Models shown in the "List of Models" are UL and ERS approved models.

Note: Note that heat-resistant models are not standard approved models.

UL (UL1054) CSA (CSA C22.2 No. 55)

Rated voltage Model	GS-T0	GS-S	GS-B1
125 VAC	-	5 A	0.1 A
250 VAC	10.1 A	3 A	-
30 VDC	-	-	0.1 A

Consult your OMRON sales representative for specific models with VDE standard approvals.

VDE (EN 51058-1)

Rated Voltage	Model	RG-10	SG-5
250 VAC		10 A	5 A

Testing conditions: 9E1 (50,000 operations)

TMS 10°C to 25°C

Characteristics

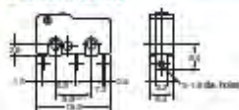
Item	Model	SD-10 models	SD-50 models	SD-40 models
Permissible operating speed		0.1 mm to 1 mm for gun plunger models		
Permissible operating frequency	Mechanical	400 operations/min		
	Electrical	10 operations/min		
Insulation resistance		100 MΩ min. (at 500 VDC with insulation tester)		
Contact resistance (initial value)	OP 1.47 N models	30 mΩ max.		50 mΩ max.
	OP 6.40 N models	-	50 mΩ max.	100 mΩ max.
	OP 6.25 N models			110 mΩ max.
Dielectric strength (1)	Between terminals of the same polarity	1,000 VAC 50/60 Hz for 1 min		
	Between current-carrying metal parts and ground	1,500 VAC 50/60 Hz for 1 min		
	Between each terminal and non-current-carrying metal parts	1,500 VAC 50/60 Hz for 1 min		
Vibration resistance (2)	Multifunction	10 to 55 Hz 1.5 mm double amplitude		
Shock resistance	Durability	1,000 mil (approx. 100 G) max.		
	OP 6.40 N models	500 mil (approx. 50 G) max.		
	OP 6.25 N models	500 mil (approx. 50 G) max.		
	Durability (3)	100 mil (approx. 10 G) max.		
	OP 6.40 N models	50 mil (approx. 5 G) max.		
	OP 6.25 N models	50 mil (approx. 5 G) max.		
Durability (4)	Mechanical	10,000,000 operations min. (10 operations/min)	30,000,000 operations min. (10 operations/min)	
	Electrical	10,000 operations min. (10 operations/min)	300,000 operations min. (10 operations/min)	
Degree of protection		IP67 (IP68)		
Degree of protection against electric shock		Class I		
Proof tracking index (PTI)		1		
Ambient operating temperature		-25°C to +55°C (at ambient humidity of 95% max.) (with no icing or condensation)		
Ambient operating humidity		85% max. (for -10°C to +55°C)		
Weight		Approx. 1.5 kg (gun plunger model)		

Note. The data given above are initial values.

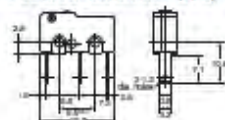
- *3. For testing conditions, consult your OMRON sales representative.

Terminals/Appearances (Unit: n n)

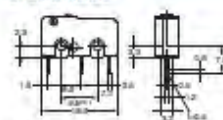
- Solder terminals



- Quick connect terminals (\$110)



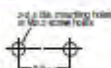
- PCB terminals



PCD Mounting Dimensions (Reference)



Mounting Holes (Unit: mm)



Dimensions (Unit: mm) and Operating Characteristics

The illustrations and drawings are for solder terminal models.

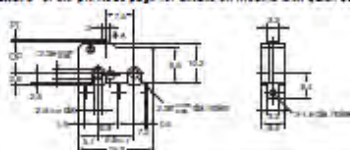
Refer to "Terminals/Appearance" of the previous page for details on models with quick connect terminals (W110) or PCB terminals.

●Pin plunger

SS-10

SS-5 (-F)

SS-01 (-E, -F)



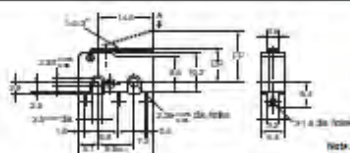
Operating Characteristics	Model	SS-10	SS-5	SS-5-F	SS-01-E
Operating Force	OP	Max. 1.47 N (150 gf)	1.47 N (150 gf)	0.49 N (50 gf)	0.20 N (20 gf)
Releasing Force	RF	Min. 0.29 N (30 gf)	0.20 N (20 gf)	0.04 N (4 gf)	0.00 N (0 gf)
Overtravel	OT	Max. 0.6 mm	0.6 mm	0.6 mm	0.6 mm
Overtravel	OT	Min. 0.4 mm	0.5 mm	0.5 mm	0.5 mm
Movement Differential	MD	Max. 0.12 mm	0.1 mm	0.1 mm	0.1 mm
Operating Position	OP	0.1±0.5 mm			

●Hinge lever

SS-10GL

SS-5GL (-F)

SS-01GL (-E, -F)



Note: The indicated reference values of RF are for cases where the lever weight is not applied to the plunger.

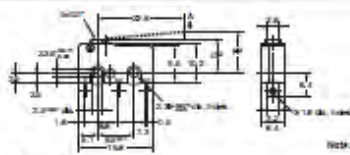
Operating Characteristics	Model	SS-10GL	SS-5GL	SS-5GL-F	SS-01GL-E
Operating Force	OP	Max. 0.49 N (50 gf)	0.49 N (50 gf)	0.10 N (10 gf)	0.00 N (0 gf)
Releasing Force	RF	Min. 0.00 N (0 gf)	0.00 N (0 gf)	0.00 N (0 gf)	0.01 N (1 gf) (reference value)
Overtravel	OT	Min. 1.0 mm	1.2 mm	1.2 mm	1.2 mm
Movement Differential	MD	Max. 1.0 mm	0.6 mm	0.6 mm	0.6 mm
Free Position	FR	15.0 mm			
Operating Position	OP	0.1±0.5 mm			

●Long hinge lever

SS-10GL111

SS-5GL111 (-F)

SS-01GL111 (-E, -F)



Note: The indicated reference values of RF are for cases where the lever weight is not applied to the plunger.

Operating Characteristics	Model	SS-10GL111	SS-5GL111-F	SS-01GL111-E	SS-01GL111-E
Operating Force	OP	Max. 0.30 N (30 gf)	0.10 N (10 gf)	0.00 N (0 gf)	0.00 N (0 gf)
Releasing Force	RF	Min. 0.00 N (0 gf)	0.00 N (0 gf)	0.00 N (0 gf)	0.00 N (0 gf)
Overtravel	OT	Min. 1.2 mm	1.2 mm	1.2 mm	1.2 mm
Movement Differential	MD	Max. 1.2 mm	0.6 mm	0.6 mm	0.6 mm
Free Position	FR	15.0 mm			
Operating Position	OP	0.1±0.5 mm			

Note 1: Unless otherwise specified, a tolerance of 0.1 mm applies to all dimensions.

Note 2: The operating characteristics are for operation in the A direction (↓).

5. Datasheet Reed Switch

ORD211-37-21

REED SWITCH

ORD211

General Purpose Ultraminiature (Low-level Load 24 V Max.)

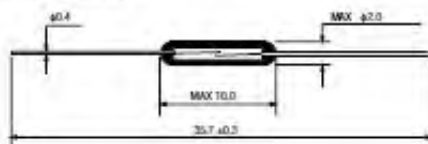
GENERAL DESCRIPTION

The ORD211 is a small single-contact reed switch designed for general control of low-level loads less than 24 V. The contacts are sealed within the glass tube with inert gas to maintain contact reliability.

Features

- (1) Reed contacts are hermetically sealed within a glass tube with inert gas and do not receive any influence from the external atmospheric environment.
- (2) Quick response
- (3) The structure comprises an operating system and electrical circuits coaxially. Reed switches are suited to applications in radio frequency.
- (4) Reed switches are compact and light weight.
- (5) Superior corrosion resistance and wear resistance of the contacts assures stable switching operation and long life.
- (6) With a permanent magnet installed, reed switches economically and easily become proximity switches.

External Dimensions (Unit:mm)



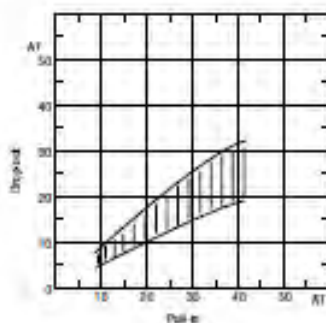
APPLICATIONS OF REED SWITCHES

1. Automotive electronic devices
2. Control equipment
3. Communication equipment
4. Measurement equipment
5. Household appliances

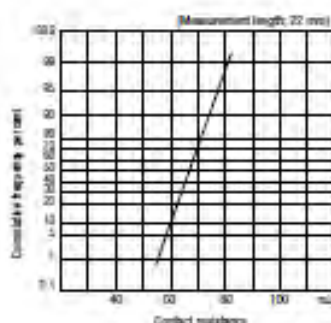
ELECTRICAL CHARACTERISTICS

Parameter	Symbol	Condition	Rated Value			Unit
			Min.	Typ.	Max.	
Pull-In Value	PI	—	10	—	40	AT
Drop-out Value	DO	—	5	—	—	AT
Contact Resistance	CR	—	—	—	100	mΩ
Breakdown Voltage	—	—	100	—	—	VDC
Insulation Resistance	—	—	10^9	—	—	Ω
Electrostatic Capacitance	—	—	—	—	0.2	pF
Contact Rating	—	—	—	—	1.0	VA
Maximum Switching Voltage	—	—	—	—	24	V
Maximum Switching Current	—	—	—	—	0.1	A
Maximum Carry Current	—	—	—	—	0.3	A

(1) Drop-out vs. Pull-in



(2) Contact resistance



DAFTAR RIWAYAT HIDUP



Nama : Shofia Kholisatus Sa'adah
TTL : Kediri, 24 Januari 1994
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Jalan Kyai Doko No. 230
Desa Doko, Kec. Ngasem,
Kab. Kediri
Telp/HP : 085646450650
E-mail : shofiakholisatus@gmail.com

RIWAYAT PENDIDIKAN

1. 2000 – 2006 : SD Negeri Banjaran V Kediri
2. 2006 – 2009 : SMP Negeri 1 Kediri
3. 2009 – 2012 : SMA Negeri 2 Kediri
4. 2012 – 2015 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PLTA Selorejo Ngantang, Malang, Jawa Timur
2. Kerja Praktek di PG. Pesantren Kediri

PENGALAMAN ORGANISASI

- | | | | | | | |
|----|--------|------------|-------|--------|---------|-------------------------|
| 1. | Staff | Departemen | Dalam | Negeri | Periode | 2013/2014 |
| | | | | | | HIMAD3TEKTRO, FTI - ITS |
| 2. | Kabiro | Departemen | Dalam | Negeri | Periode | 2014/2015 |
| | | | | | | HIMAD3TEKTRO, FTI - ITS |

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Alin Wahyu Setiawan
TTL : Samarinda, 07 Nopember 1994
Jenis Kelamin : Laki - Laki
Agama : Islam
Alamat : Jalan P. Diponegoro Gg.
Darmawan, Desa Bukuan,
Kec. Palaran, Samarinda
Telp/HP : 085250077040
E-mail : alinwahyus@gmail.com

RIWAYAT PENDIDIKAN

1. 2000 – 2006 : SD Negeri 002 Palaran
2. 2006 – 2009 : SMP Negeri 14 Samarinda
3. 2009 – 2012 : SMA Negeri 6 Samarinda
4. 2012 – 2015 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Hidup Baru Perdana Abadi, Balikpapan
2. Kerja Praktek di PT. Telkomsel Indonesia
3. Kerja Praktek di Laboratorium Mekanika Pusat Robotika ITS

PENGALAMAN ORGANISASI

1. Anggota Staff Himpunan Mahasiswa Islam Sepuluh Nopember
2. Anggota Forum Komunikasi Mahasiswa Programmer Indonesia

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

Gambar 2.1	Brankas	6
Gambar 2.2	Mikrokontroler ATmega 128	6
Gambar 2.3	Contoh Pola Keluaran <i>Incremental Encoder</i>	8
Gambar 2.4	Modem GSM SIM 900A	9
Gambar 2.5	DC Planet <i>Geared</i> Motor	10
Gambar 2.6	DC Motor <i>Driver</i>	11
Gambar 2.7	LCD 8 x 2	12
Gambar 2.8	Reed Switch	12
Gambar 2.9	Limit Switch	13
Gambar 3.1	Blok Fungsional Sistem	14
Gambar 3.2	Skematik Mikrokontroler ATmega 128	16
Gambar 3.3	Skematik Modul GSM SIM 900A	18
Gambar 3.4	<i>Wiring</i> Pengkoneksian SIM 900A dengan Kontroler	18
Gambar 3.5	<i>Wiring</i> Konfigurasi <i>Driver</i> Motor	19
Gambar 3.6	<i>Voltage Regulator</i> IC LM2596	20
Gambar 3.7	Skematik <i>Voltage Regulator IC LM2596</i>	20
Gambar 3.8	<i>Wiring</i> Sensor <i>Limit Switch</i> dan Sensor <i>Reed Switch</i>	21
Gambar 3.9	Robot Pembuka Brankas	22
Gambar 3.10	Pemegang <i>Knob</i>	22
Gambar 3.11	Peluru pada Pemegang <i>Knob</i>	23
Gambar 3.12	<i>Box Acrylic</i> Elektronik	24
Gambar 3.13	<i>Box Acrylic</i> Sisi Depan	25
Gambar 3.14	<i>Box Acrylic</i> Pelindung Motor	26
Gambar 3.15	<i>Box Acrylic</i> Keseluruhan	26
Gambar 3.16	Perancangan Dudukan Motor	27
Gambar 3.17	Batangan Aluminium Penyangga	28
Gambar 3.18	Pegangan Robot untuk <i>Operator</i>	28
Gambar 3.19	Penghisap pada Lengan Robot	29
Gambar 3.20	<i>Holder</i> pada Lengan Robot	30
Gambar 3.21	<i>Space</i> Penggerak pada Lengan Robot	31
Gambar 3.22	Batangan Aluminium Penghubung	32
Gambar 3.23	Kaki Penyangga Robot	33
Gambar 3.24	Alas Aluminium Sensor <i>Limit Switch</i>	34
Gambar 3.25	<i>Flowchart</i> Program Keseluruhan	35
Gambar 3.26	Konfigurasi <i>Code Wizard</i> AVR	39
Gambar 4.1	<i>Wiring</i> Diagram Pengkoneksian	41

Gambar 4.2	Tampilan pada <i>Hyperterminal</i>	42
Gambar 4.3	Modul <i>Driver</i> Motor	43
Gambar 4.4	<i>Flowchart</i> Program Pembacaan Rotary Encoder	45
Gambar 4.5	Grafik Pengujian Program Pembacaan <i>Rotary Encoder</i>	47
Gambar 4.6	<i>Flowchart</i> Program <i>Driver</i> Motor	48
Gambar 4.7	<i>Flowchart</i> Program Modul GSM SIM 900A	51
Gambar 4.8	Pengujian Program Modul GSM SIM 900A (1)	52
Gambar 4.9	Pengujian Program Modul GSM SIM 900A (2)	52
Gambar 4.10	Pengujian Kepresisian Motor	55
Gambar 4.11	Memasang Robot pada Pintu Brankas	56
Gambar 4.12	Memposisikan <i>Knob</i> pada Posisi 0	56
Gambar 4.13	Menutup Bagian <i>Security</i> pada Brankas	57
Gambar 4.14	Mengaktifkan <i>Push Button Power</i>	57
Gambar 4.15	Mengaktifkan <i>Push Button Start</i>	58
Gambar 4.16	Brankas Sudah Terbuka	58
Gambar 4.17	Brankas Terbuka dan <i>User</i> Menerima SMS Konfirmasi	59
Gambar 4.18	Pengujian Secara Keseluruhan	60

TUGAS AKHIR – TE 145561

RANCANG BANGUN ROBOT PEMBUKA BRANKAS DENGAN SISTEM PENGIRIMAN KODE PUTAR MELALUI SMS *GATEWAY*

Oleh :

Shofia Kholisatus Sa'adah 2212039002

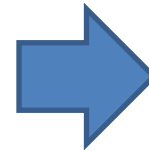
Alin Wahyu Setiawan 2212039043

Dosen Pembimbing :

Rudy Dikairono, S.T., M.T

Onie Meiyanto, S.Pd

LATAR BELAKANG



Robot Pembuka Kunci
Brankas dengan Pengiriman
Kode Putar Melalui SMS
Gateway

PERMASALAHAN

Permasalahan yang dibahas dalam Tugas Akhir ini adalah sebagai berikut :

Tidak ada teknologi yang dapat membuka kunci brankas manual pada jarak jauh melalui SMS

Pemilik brankas tidak selalu *standby* untuk membuka kunci brankas saat ada aset yang dibutuhkan

BATASAN MASALAH

- Robot ini dapat membuka kunci brankas dengan brankas *type* BERNSTEIN size 2
- Tugas Akhir ini hanya dapat membuka kunci brankas dengan kode putar yang dikirim melalui SMS.
- Pengontrolan robot ini menggunakan mikrokontroler ATmega 128 dan Modul GSM SIM 900A.
- Sensor yang digunakan adalah sensor rotary encoder yang terdapat pada DC Planet *Geared* Motor Type PG36 itu sendiri.
- Selain itu menggunakan sensor *reed switch* dan *limit switch* pula.

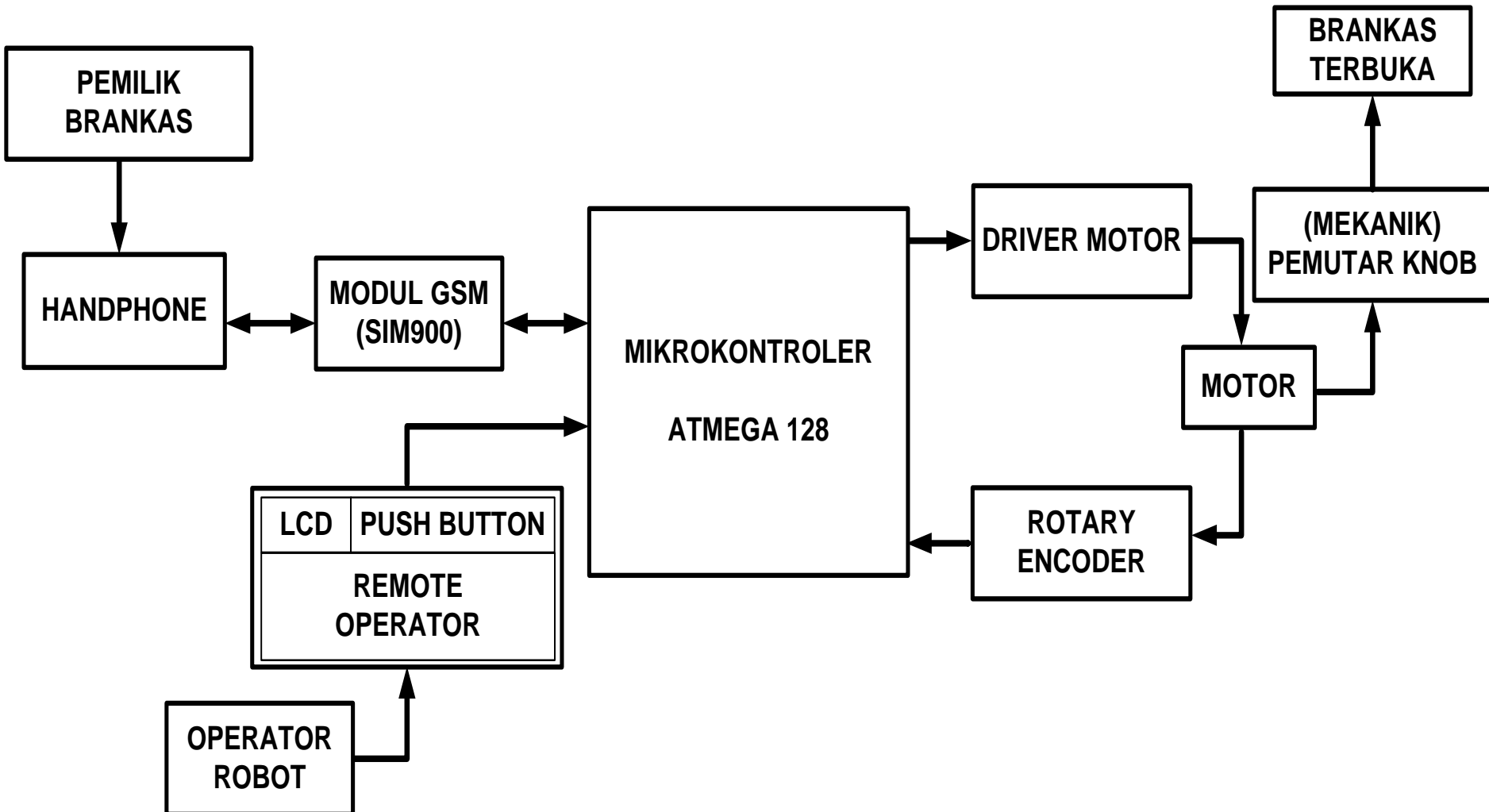
TUJUAN

Tujuan yang ingin dicapai dari pembuatan Tugas Akhir ini adalah sebagai berikut :

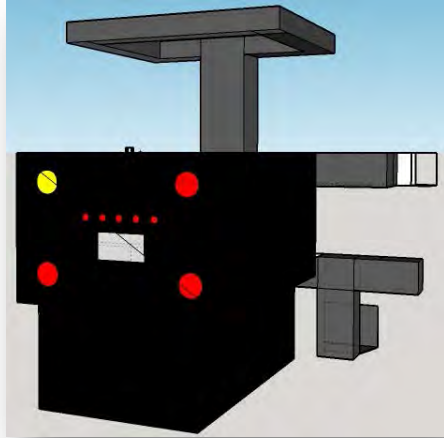
Merancang dan membuat robot yang mampu membuka kunci brankas manual yang mana kode brankas dikirim melalui SMS

Memberi kemudahan kepada pemilik brankas untuk membuka brankas yang dapat dilakukan dari jarak jauh dengan aman

BLOK DIAGRAM SISTEM

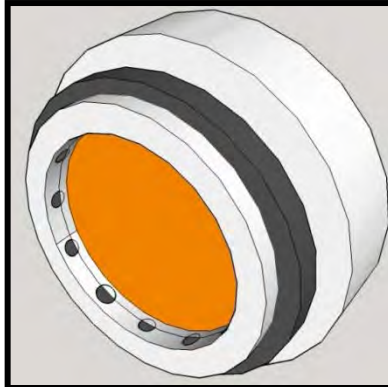
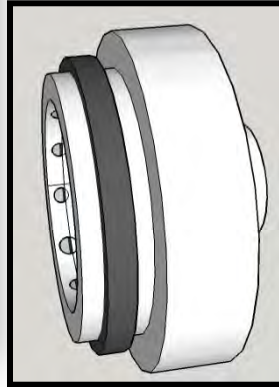


PERANCANGAN MEKANIK

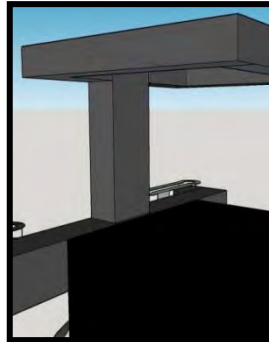
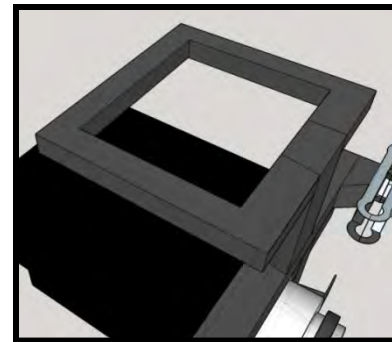


Perancangan Secara Keseluruhan

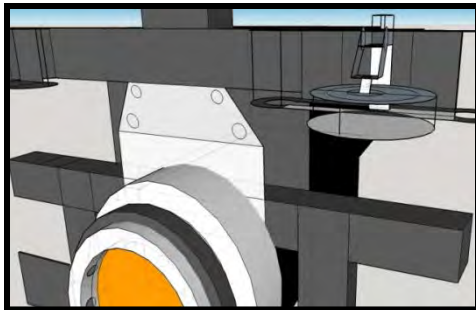
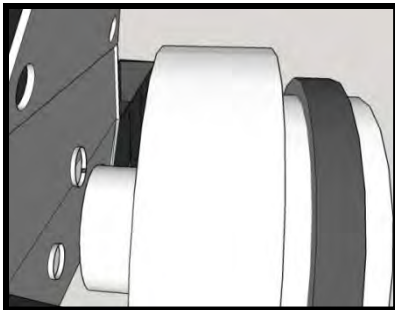
Perancangan pemegang *knob*



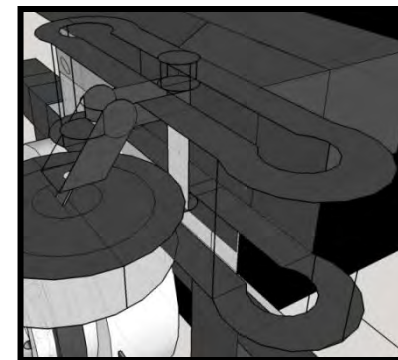
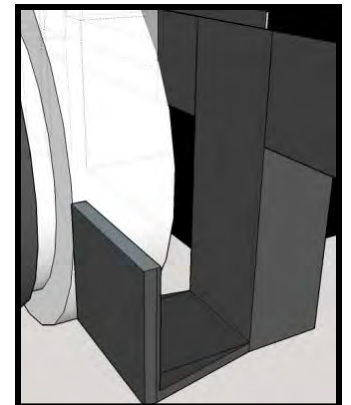
Perancangan pegangan operator



Perancangan dudukan motor

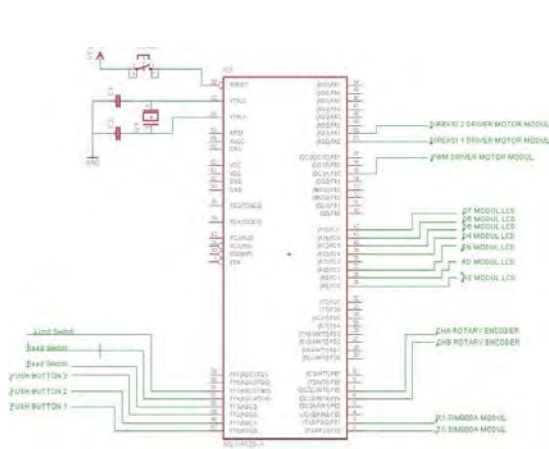


Perancangan kaki robot

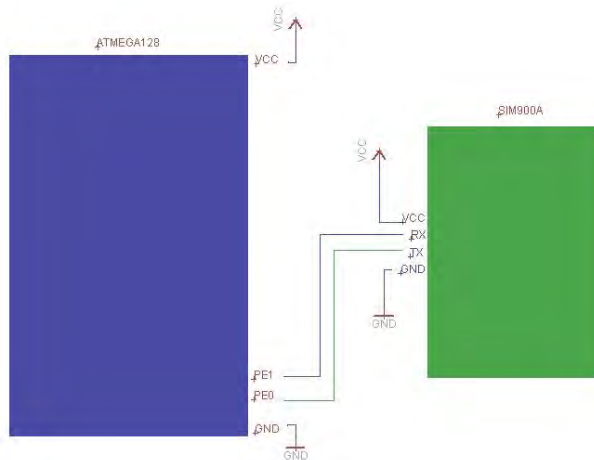


Perancangan lengan robot

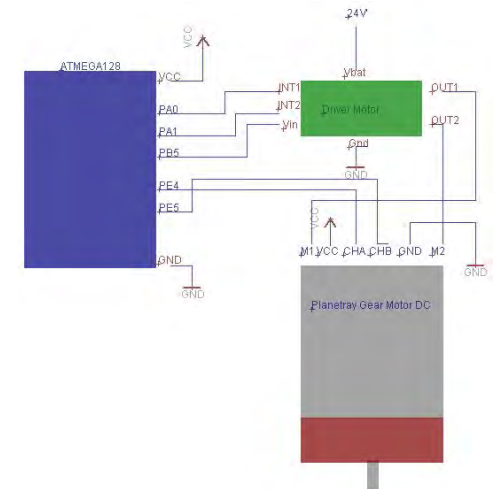
PERANCANGAN ELEKTRIK



**SKEMATIK MIKROKONTROLER
ATMEGA 128**



**PENGKONEKSIAN MODUL GSM
SIM 900A**

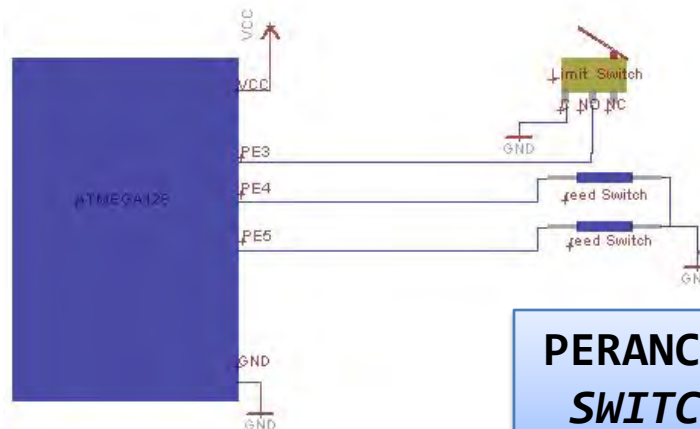


**PENGENDALIAN DC
PLANET GEARED
MOTOR MENGGUNAKAN
DRIVER MOTOR**



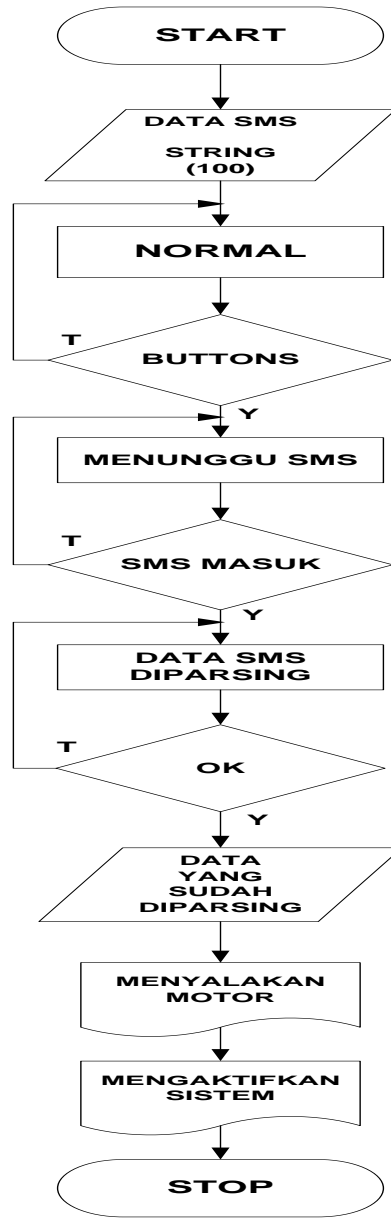
IC LM2596

**PERANCANGAN VOLTAGE
REGULATOR**

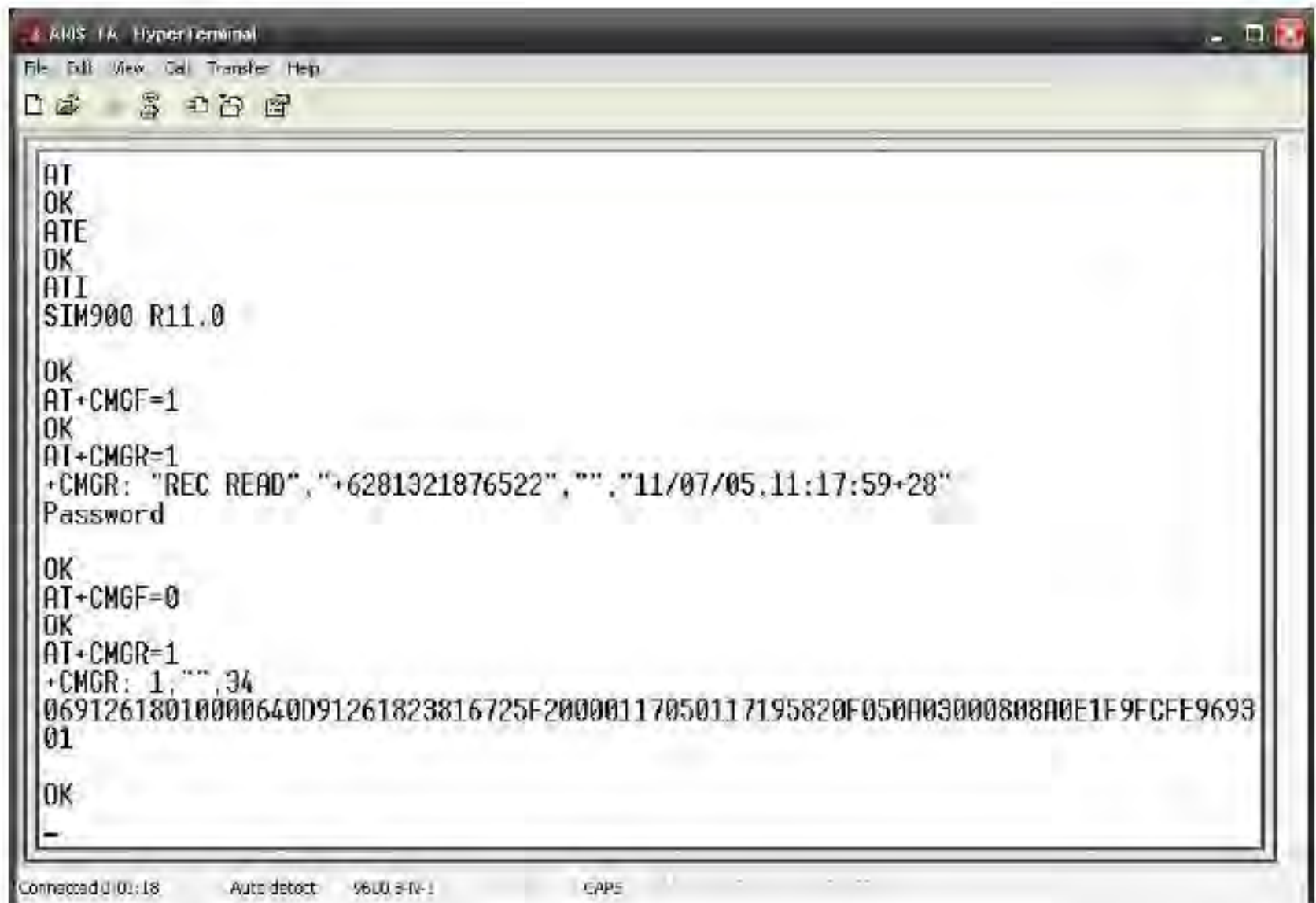


**PERANCANGAN SENSOR *LIMIT*
SWITCH DAN REED SWITCH**

FLOWCHART PROGRAM



PENGUJIAN MODUL GSM SIM 900A



The screenshot shows a HyperTerminal window titled "ANS 1A HyperTerminal". The window contains a series of AT commands and their responses. The commands are: AT, OK, ATE, OK, AT+SIM900, SIM900 R11.0, OK, AT+CMGF=1, OK, AT+CMGR=1, +CMGR: "REC READ", "+6281321876522", "", "11/07/05,11:17:59+28", Password, OK, AT+CMGF=0, OK, AT+CMGR=1, +CMGR: 1, "", 34, 06912618010000640D91261823816725F20000117050117195820F050A03000808A0E1F9FCFE9693, 01, OK, and a final dash (-). The status bar at the bottom indicates "Connected 01/01/18", "Auto detect", "9600, 8-N-1", and "CPS".

```
ANS 1A HyperTerminal
File Edit View Call Transfer Help
[Icons]

AT
OK
ATE
OK
AT+SIM900
SIM900 R11.0

OK
AT+CMGF=1
OK
AT+CMGR=1
+CMGR: "REC READ", "+6281321876522", "", "11/07/05,11:17:59+28"
Password

OK
AT+CMGF=0
OK
AT+CMGR=1
+CMGR: 1, "", 34
06912618010000640D91261823816725F20000117050117195820F050A03000808A0E1F9FCFE9693
01

OK
-
```

Connected 01/01/18 Auto detect 9600, 8-N-1 CPS

PENGUJIAN DC *PLANETARY GEARED* MOTOR

M1	M2	Kondisi Motor
24,6	Gnd	Berputar CW
Gnd	24,6	Berputar CCW
Gnd	Gnd	Motor tidak Berputar

Keterangan :

CW = Clock Wise (searah jarum jam)

CCW = Counter Clock Wise (berlawanan arah jarum jam)

PENGUJIAN *DRIVER MOTOR*

No.	INT 1	INT 2	Vin	Vout	Kondisi Motor
1.	0V	0V	0V	24,9V	Diam
2.	0V	0V	5V	24,9V	Diam
3.	0V	5V	0V	24,9V	Diam
4.	5V	0V	0V	24,9V	Diam
5.	0V	5V	5V	24,9V	Berputar Forward
6.	5V	0V	5V	24,9V	Berputar Reverse
7.	5V	5V	0V	24,9V	Diam
8.	5V	5V	5V	24,9V	Diam

PENGUJIAN PROGRAM *DRIVER MOTOR* TERHADAP DC *PLANETARY GEARED MOTOR*

Data yang Diinputkan pada Program	Data Rotary yang Terbaca	Nilai Rotary yang Diinginkan (Setpoint)	Error(Selisih)
100	754	753	-1
220	1657	1657	0
300	2260	2259	+1
440	3314	3314	0
500	3766	3766	0

PENGUJIAN PROGRAM MODUL GSM SIM 900A

Data Yang dikirim melalui Hand Phone	Data Yang diterima dan ditampilkan ke LCD			
	DATA 1	DATA 2	DATA 3	DATA 4
*00-00-10-30#OK	00	00	10	30
*12-00-12-33#OK	12	00	12	33
*22-12-10-30#OK	22	12	10	30
*12-00-10-30#OK	12	00	10	30
*23-00-00-30#OK	23	00	00	30
*11-22-33-44#OK	11	22	33	44
*123-00-10-303#OK	12	-0	-1	-3

PENGUJIAN KEPRESISIAN PUTARAN MOTOR

Data yang Diinputkan	Index Knob Awal	Rotary yang Dibaca di LCD	Index Knob Setelah Motor Berjalan	Kondisi Motor
100	0	754	0	Motor CCW berputar satu kali putaran
-300	0	-2260	0	Motor CW berputar 3 kali putaran
200	0	1507	0	Motor CCW berputar 2 kali putaran
400	0	3013	0	Motor CCW berputar 4 kali putaran
600	0	4520	0	Motor CCW berputar 6 kali putaran
-50	0	-376	50	Motor CW berputar 1/2 putaran
40	0	302	40	Motor CW berputar 2/5 putaran

PENGUJIAN SECARA KESELURUHAN

Parameter	Status
<i>Pushbutton Power</i>	Mengaktifkan baterai dan sistem pada robot pembuka brankas
LED 5 buah	Indikator sistem sudah ready untuk digunakan
Push Button I (Reset)	Menghentikan sistem sehingga sistem akan refresh.
Push Button II (Start SMS)	Sistem sudah ready untuk menerima SMS dari user
Limit Switch	Mengamankan security kode
Reed Switch 1 (S1)	Mendeteksi medan magnet
Reed Switch 2 (S2)	Mendeteksi medan magnet
Modul GSM SIM 900A	Menerima SMS kode putar dari user dan mengirim SMS konfirmasi kepada user

VIDEO

KESIMPULAN

- Ketika sistem terselesaikan dengan baik maka robot akan mengirimkan SMS konfirmasi kepada *user*. Dan apabila terdapat kesalahan sistem, maka sistem akan berhenti, dan robot akan mengirim SMS konfirmasi kepada *user* bahwa sistem tidak terselesaikan.
- Brankas dapat dibuka apabila nilai putaran motor sudah sesuai dengan kode yang dikirim oleh *user*.
- Kepresisian nilai *rotary encoder* tergantung dari nilai slip dari *knob*. Semakin kecil nilai slip dari *knob* maka semakin presisi nilai *rotary encoder* nya.
- PWM (*Pulse width Modulation*) mempengaruhi kecepatan putar pada motor. Semakin besar nilai PWM nya, semakin cepat pula putaran motornya.

SARAN

- Penggunaan LCD pada robot pembuka brankas ini seharusnya ditanggalkan. Agar lebih *secure*. Sehingga informasi yang ditampilkan dapat dibatasi.

TERIMAKASIH